

SMART GRID DEMAND RESPONSE: A WIRELESS SENSOR NETWORK
APPROACH

A Dissertation
Presented to the Faculty of the Graduate School
of Cornell University
In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by
Coalton Bennett
May 2010

© 2010 Coalton Bennett
ALL RIGHTS RESERVED

SMART GRID DEMAND RESPONSE: A WIRELESS SENSOR NETWORK APPROACH

Coalton Bennett, Ph.D.

Cornell University 2010

Wireless Sensor Networks are being used now, in more applications than ever imagined. Much like the Internet and other revolutionary telecommunications technologies that have been developed in the past two decades, wireless sensor networks are slowly starting to become a part of the general populations everyday life. So much so that they are now being integrated into everyday consumer electronics and appliances, to monitor our electricity consumption habits. The goal of course is to educate customers about wasteful spending habits, and hopefully encourage them to participate in programs that will not only reduce electricity bills, but also help preserve an already archaic electric power system. To this end, customer appliances and their electric power meters must be retrofitted with small easy to use wireless sensors to form a network capable of providing real time data to the customer about their electricity consumption.

This thesis details how such a network can be created not only between appliances within a home and the customer's meter, but also how the meters themselves form a network that can be used to relay customer data to the utility. In particular we have shown how a novel current sensor design can be used to not only measure the power of appliances, but also scavenge enough energy to power the device indefinitely.

The second part of this thesis examines the Zigbee protocol, which has become an industry standard for ad-hoc smart meter networks. We exploit the reactive nature of the protocol to create loop free redundant paths between each source node (smart meter) and the sink node (utility access point), to facilitate the use of the well-known Multiprotocol Label Switching architecture with the Zigbee protocol.

BIOGRAPHICAL SKETCH

Coalton Bennett was born to Charlie Bennett and Pamela Merritt-Bennett on May 19th, 1983 in Merced California. As the son of two educators, an emphasis on education was stressed at an early age in his life. Although his mother, a biologist and his high school biology teacher, tried relentlessly to encourage her son to pursue a career in medicine, he found the study of living organisms to be mundane. Thus began his interest in the next best subject: mathematics. After high school Coalton attended Howard University in the fall of 2000 and majored in Electrical Engineering and minored in mathematics. He was introduced to the wonderful world of signal processing and wireless communications by two very well respected professors in their fields and decided to follow in their footsteps. In the fall of 2005 he was admitted to the Ph.D program in the field of Electrical and Computer Engineering. He took full advantage of the interdisciplinary research opportunities, as encouraged by his advisor, and also obtained a minor in Economics.

I would like to dedicate this work to my parents, Charlie and Pamela Bennett. Without their support I would not have accomplished what I did today.

ACKNOWLEDGMENTS

I would first like to acknowledge my advisor, who has been supportive of my ideas, and who also allowed me to develop and hone my creative abilities. He was instrumental in providing me with essential recommendations about my research that has resulted in a multitude of research opportunities post graduation. To him I am forever indebted.

I would also like to acknowledge committee members Dr. Clifford Pollock and Dr. Timothy Mount for their suggestions and contributions. Dr. Michael Spencer has also been an invaluable resource, and provided support in ways that only he can, and I am appreciative of his support.

Dr. Cardell was instrumental in providing me with a great deal guidance when I first joined the WISL group. She had taken a sabbatical that year to work with a group of power systems engineers at Cornell and was introduced to my advisor. Thus set the wheels in motion for a wonderful working relationship, that resulted in my first publication. I thank her, along with Dr. Wicker, for steering me in the right direction. Their foresight is what made it possible for me to establish meaningful relationships with industry specialists. My first and only internship, during graduate school, would not have been possible if it were not for Dr. Cardell. So to her I would like to say I am extremely grateful.

I hope that after the completion of this program we will continue to have the same working relationship that we once did when I was a student of theirs.

TABLE OF CONTENTS

| | |
|--|--------------|
| Biographical Sketch | iii |
| Dedication | iv |
| Acknowledgments | v |
| Table of Contents | vi |
| List of Figures | viii |
| List of Tables | x |
| 1 Introduction | 1 |
| 1.1 Wireless Sensor Networks | 1 |
| 1.2 Sensor Networks an Application in Electric Power Systems..... | 3 |
| 2 Home Area Network | 6 |
| 2.1 Home Area Network Introduction | 6 |
| 2.1.1 List of HAN Device Communication Options | 8 |
| 2.1.1.1 Zigbee Home Automation Designs | 13 |
| 2.1.1.2 Sensor Design | 15 |
| 2.1.2 Piezoelectricity | 18 |
| 2.1.2.1 Direct Piezoelectric Effect | 20 |
| 2.1.2.2 Converse Effect | 22 |
| 2.1.2.3 Direct, Converse Piezoelectric, and Constitutive Equations | 22 |
| 2.1.2.4 Electric Field and Electric Displacement | 23 |
| 2.1.2.5 Stress and Strain | 24 |
| 2.1.2.6 Strain and Electric Field | 24 |
| 2.1.2.7 Stress and Electric Displacement | 25 |
| 2.1.2.8 Measured voltage as a function of tip deflection | 28 |
| 2.2 Sensor Design | 31 |

| | |
|--|------------|
| 2.2.1 Current as an Instrument for Power | 33 |
| 2.3 Zigbee and 802.15.4 Overview | 39 |
| 2.4 Overall System Design | 40 |
| 3 Neighborhood Area Network | 41 |
| 3.1 Neighborhood Area Network Introduction | 41 |
| 3.1.1 AODV | 41 |
| 3.1.1.2 Route Discovery | 44 |
| 3.1.1.3 Path Maintenance | 48 |
| 3.1.1.4 Local Connectivity | 49 |
| 3.1.1.5 Local Repair | 52 |
| 3.1.2 AODV & Zigbee | 53 |
| 3.2 Smart Meters & Demand Response Systems | 56 |
| 3.2.1 Smart Meters | 62 |
| 3.2.1.1 Time Intolerant Wireless Sensor Networks | 63 |
| 3.3 AODV-MPLS Protocol | 69 |
| 3.3.1 Route Setup and Label Distribution | 70 |
| 3.3.2 Routing Table Entries | 70 |
| 3.3.3 Timing | 75 |
| 3.3.4 Forward Equivalence Class | 76 |
| 3.3.5 Labels | 78 |
| 3.3.6 Forward Path (Toward Sink) | 79 |
| 3.3.7 Reverse Path (Toward Source) | 84 |
| 3.4 Results and Analysis | 91 |
| 4 Looking Forward & Conclusion | 99 |
| Bibliography | 102 |

LIST OF FIGURES

| | | |
|------|--|----|
| 2.1 | Frequency Response of Appliances | 10 |
| 2.2 | Powerline Carrier Transceiver | 12 |
| 2.3 | Unit Cube of Piezo Material | 20 |
| 2.4 | Piezoelectric Effect | 22 |
| 2.5 | Cantilevered Beam | 29 |
| 2.6 | Current Sensor coupled with MICAz | 33 |
| 2.7 | Current Sensor coupled with MICAz | 33 |
| 2.8 | Power vs Time for Sample 2 | 37 |
| 2.9 | Power vs Time for Sample 3 | 38 |
| 2.10 | Power vs Time for Sample 1 | 38 |
| 3.1 | Example Cluster Tree Network | 44 |
| 3.2 | Route Request Message Format | 46 |
| 3.3 | Route Reply Message Format | 48 |
| 3.4 | Route Error Message Format | 50 |
| 3.5 | Legacy Power System/Grid | 58 |
| 3.6 | Potential Smart Grid Illustration | 58 |
| 3.7 | Advanced Metering Infrastructure Interfaces | 60 |
| 3.8 | Chain of Nodes | 64 |
| 3.9 | General Network Frame Format | 74 |
| 3.10 | Frame Control Field | 75 |
| 3.11 | Values for Frame Type Sub-Field | 75 |
| 3.12 | Zigbee Route Request Command Frame | 75 |
| 3.13 | AODV-MPLS Zigbee Route Request Command Frame | 75 |
| 3.14 | Zigbee Route Reply Command Frame | 77 |

| | | |
|------|--|----|
| 3.15 | AODV-MPLS Route Reply Command Frame | 77 |
| 3.16 | Source Route Subfame | 80 |
| 3.17 | Example Network | 82 |
| 3.18 | AODV-MPLS Label and Experimental Field | 90 |
| 3.19 | Delay vs Time – 48 hour period | 92 |
| 3.20 | 100 Node Network | 93 |
| 3.21 | Throughput vs Time – 48 period | 94 |
| 3.22 | Number of IP Hops for AODV and AODV-MPLS | 94 |
| 3.23 | Number of Dropped Packets for AODV and AODV-MPLS | 95 |
| 3.25 | Route Discovery Time | 96 |
| 3.26 | Received Routing Traffic | 97 |
| 3.27 | Routing Traffic Sent | 97 |
| 3.28 | Total Cached Replies Sent | 98 |
| 3.29 | Number of Replies Sent from Destination | 98 |

LIST OF TABLES

| | | |
|-----|--|----|
| 2.1 | Home Area Network Requirements | 6 |
| 2.2 | Home Automation Protocols | 12 |
| 2.3 | Linear Regression Coefficients | 37 |
| 3.1 | Example Network Legend | 44 |
| 3.2 | AODV-MPLS Route Table Entry | 72 |
| 3.3 | Route List | 72 |
| 3.4 | Precursor List | 72 |
| 3.5 | Zigbee Route Table Entry | 73 |
| 3.6 | Zigbee Route Discovery Table Entry | 74 |
| 3.7 | Global Network Time | 90 |

CHAPTER 1

INTRODUCTION

1.1 Wireless Sensor Networks

Wireless sensor networks have recently began to peak the interest of researchers and industry specialists in a variety of different fields of engineering. Hewlett-Packard has launched a campaign to create a “Central Nervous System of the Earth” [1] by placing literally 1 trillion sensors across the globe to collect meaningful information about the Earth’s environment that could be useful for researchers and industry specialists alike. Currently wireless sensor networks are used quite regularly in: ecological habitat monitoring [2], structure health monitoring [3], environmental contamination detection [4], industrial process control [5], and military tracking [6].

The earliest known use of sensor networks dates back to the 1970s, albeit the devices were relatively small in scale they were not wireless. In the early 1990s the idea of large scale embedded systems became a reality when low power VLSI chipset components became easier to produce. One of the very first unclassified projects to be published was [7]. The objective of the project was to create a device with low-power components so that a large-scale wireless sensor network could be created with a sizeable number of sensors. Later that decade the very first wireless sensor device was created. One of the better-known wireless sensor networking devices, are called motes [8]. Motes have made it easier for researchers to conduct projects, because of the relatively simple embedded operating system design that accompanies the microprocessor and wireless radio.

A wireless sensor consists of several very important components. The first, and probably most important component, is a low-power embedded processor. Each device that belongs to a network is responsible for not only processing data that it generates itself, but must also relay information from other devices within its communication radius. Unfortunately most of the applications that researchers and industry experts would like to use embedded wireless devices for require the devices to expend a tremendous amount of energy thereby decreasing the lifetime of the deployment. Typically most networks incorporate low-power routing techniques, efficient sleep modes and schedules, and in the future possibly dynamic voltage scaling to increase the lifetime of the network. Recent advancements using Micro Electro Mechanical Systems (MEMS) devices has proven to be useful in providing an answer to this problem [9].

The second component of a wireless sensor is memory. The amount of memory available for a device to use is limited by economic factors, because sophisticated components are needed in order to provide the levels of memory that certain deployments might require.

The third component is the radio transceiver, which is usually limited in range and the amount of data that it can transmit per unit time. Each transmission that a device makes consumes a sizeable amount of the devices energy and thus making it necessary for energy aware or constraint based routing protocols to be used. Or even sleep/wake based scheduling algorithms.

The fourth component is the sensor, which usually only supports low-rate sensing. Each node might be equipped with several sensing devices capable of sensing:

temperature, light, humidity, pressure, accelerometer, magnetometer, chemical, acoustic, low-resolution imagers, and electric current.

The fifth component is the geopositioning system. In many, if not almost all applications, the time in which an event that is being sensed occurs is just as important as where it has occurred. Most wireless sensor networks are deployed in an ad hoc fashion and therefore the methods for obtaining the location of certain nodes in the network is necessary but oftentimes difficult to do. GPS is used with networks that have nodes equipped with GPS enabled transceivers, but this drives the cost up of the device. Localization algorithms have been developed to work around this issue.

The sixth and one of the more important components is power. Most devices are battery powered thus limiting the lifetime of the device. Depending on the application, some devices may be deployed for several years, whereas other applications might require frequent sensing thus changing the energy requirements for the device. Energy harvesting techniques, whereby each sensor is equipped with a mass and spring system, provide a way to increase device lifetime. Some applications have the ability to provide an infinite power source to each device. In this thesis we discuss one such application and the method used to exploit this power source to create a perpetual network.

1.2 Sensor Networks an Application in Electric Power Systems

Sensor networks have a variety of applications some of which were outlined above. Although until recent the idea of using these tiny embedded devices to monitor the electric power system has been given little consideration. An increased awareness of global warming and its potentially harmful effects, has spawned a cohort of:

engineers, economists, and policy makers who have come together to develop lasting solutions to one of the World's most pressing matters. Power system researchers and industry specialists are constantly striving to improve the three major components of the electric power system: generation, transmission, and distribution. To this end, participants in the generation industry have recently begun to dedicate a significant amount of time and energy to the study of the introduction of renewable energy sources such as: wind, solar, and hydroelectric power into the "generation mix". Transmission industry researchers have been toying with the idea of using superconductive material to reduce the losses associated with transmitting power over long distances [10]. There are a number of distribution system researchers who have generated ideas and proofs of concepts for different types of distributed generation (e.g. microgrids [11]) and solar panel arrays that sit on top of homes or buildings. Google and Walmart, just to mention a few, have begun major thrusts to use "green" energy sources [12][13]. Each of these ideas provides environmentally friendly ways to help meet the nation's growing demand for power. Another concept called demand response has also been given a considerable amount of attention as well, and some of the more popular demand response enabling technologies, are discussed in this thesis.

All of the aforementioned technologies are beneficial in their own right. When these separate technologies have the ability to communicate with one another, and are combined in an optimal way to benefit the customer and the three components of the electric power system, the electric power system of today will become what some call a 'smart' power system, or the *smart grid*.

Although a definition for smart grid does not exist yet, the Public Utilities Commission of the state of California have given a lengthy definition of what it should

and should not include to ensure “... safe, reliable, efficient, and secure electrical service, with infrastructure that can meet future growth in demand...” [14]. In particular “... cost-effective smart technologies, including real time, automated, interactive technologies that optimize the physical operation of appliances and consumer devices for metering, communications concerning grid operations and status, and distribution automation...” should be used. There are myriad of different ‘smart’ technologies that provide the functionalities required above, but only the sensor networking option provides the customer with the flexibility to reuse the sensors with different consumer devices and appliances. A low-power low data rate sensor networking protocol exists, with a suite of profiles designed especially to meet the requirements set forth in the Senate Bill cited above. In what follows is a two—part discussion about how the general requirements listed in the Senate Bill can be achieved using the low-power data rate sensor networking protocol called Zigbee. The first part provides an in depth explanation of a novel sensor design capable of calculating the power, of certain appliances with linear circuit elements, by using current as a proxy for power. In the second part of the discussion details about the components of the Zigbee routing protocol are addressed, and improvements are made by exploiting the reactive nature of the protocol to create a new protocol with decreased latency and the same average throughput.

CHAPTER 2

HOME AREA NETWORK

2.1 Home Area Network Introduction

The Home Area Network (HAN) is comprised of a collection of intelligent appliances, or as they are more commonly referred to as—smart appliances for the purpose of in-premise communication [15]. By definition a smart appliance is a HAN device, typically a white good or other household appliance, that is capable of receiving wirelessly transmitted signals from the Utility and adjusting its operational mode based on consumer preferences (e.g., energy saving mode, delayed turn on/off). The HAN device is owned by the consumer, but could potentially be controlled by the utility if the consumer enrolls in an energy savings program. By definition the HAN must provide the following services to the customer and utility in order to implement a fully functional demand response network as defined above.

Table 2. 1 Home Area Network Requirements

| | |
|-------------------------|--|
| HAN Applications | <i>Control:</i> Applications running on the HAN devices must be capable of sending and receiving control information from the ESI ¹ |
| | <i>Measurement and Monitor:</i> An application must be able to provide power measurements and ensure that the device is operating within the specified limits as designated by the demand response program |

¹ Provides security and, often coordination functions that enable secure interactions between relevant Home Area Network Devices and the Utility. Permits applications such as remote load control, monitoring, and control of distributed generation, in-home display of customer usage, reading of non-energy meters, and integration with building management systems. Also provides auditing/logging functions that record transactions to and from Home Area Network Devices.

Table 2. 1 Continued

| | |
|-----------------------|--|
| | <i>Processing:</i> Applications are responsible for processing: ESI data (external), measured data (internal), and monitoring data (internal) |
| | <i>Human Machine Interface (HMI):</i> Human intervention will be necessary when system parameters must be changed; this can be accomplished by using any of the methods described in [15] |
| Communications | <i>Discovery:</i> When a customer wants to enroll a new HAN device in the demand response program the HAN should identify the device |
| | <i>Commissioning:</i> Once the device has been identified the device should be able to join the HAN via the ESI |
| | <i>Control:</i> The device should be capable of changing device parameters (e.g. switching communication channels, contending for access to channel) |
| Security | <i>Access Control and Confidentiality:</i> Access to certain physical components is dependent on the type of data. Data not corresponding with a particular physical module on the device should not be granted access. |
| | <i>Registration and Authentication:</i> Each device which has been registered with the HAN through the ESI must be granted rights to perform certain actions within the network (e.g. a refrigerator should not be able to pose as a washing machine and begin executing washing machine instruction sets) |
| | <i>Integrity:</i> Each device should be guaranteed to provide non falsified data |

Table 2. 1 Continued

| | |
|--------------------|--|
| Performance | <i>Availability:</i> Each device and the applications running on it should be accessible by the ESI at all times |
| | <i>Reliability:</i> The applications on the device should be capable of performing the tasks necessary for proper operation |
| | <i>Maintainability:</i> The ESI should be able to manage and diagnose a problem should one manifest itself in the application. |
| | <i>Scalability:</i> The network should allow, and compensate for, any number of devices attempting to join the network. However the trust center located in the ESI should be able to discern legitimate devices attempting to join and malicious agents |
| | <i>Upgradeability:</i> Over the air programming should be available on all appliances within the HAN, allowing devices to be reprogrammed or upgraded as needed |

Each of these requirements are general enough to be implemented in a number of different ways thus allowing utilities, vendors, and customers to choose the approach of their choice. Although this leaves room for innovation, it will be shown that the best design option for each HAN device consists of a: wireless radio with a microcontroller, a relay, and a current sensor.

2.1.1 List of HAN Device Communication Options

Several communication options are available for the transmission of data between the HAN device and the ESI. Traditional coax cable or Ethernet twisted pair wiring is an option, power line carrier has become an attractive alternative, and the RF option is probably the most preferred. Each of these options has advantages and disadvantages.

The Ethernet and coaxial cable option is attractive to customers who already have existing infrastructure in place via an Internet service provider. For customers who do not have an Internet service provider, the cost associated with installing new cables and wiring will exceed customer savings. Even for those customer who do have existing subscriptions with an Internet service provider, the service provider will be inclined to charge the utility to use its infrastructure for a demand response program. This cost will eventually be passed along to the customer resulting in overhead charges that could be eliminated if alternate option is selected.

Power line carrier is an attractive option, because all communication signaling is done through the wiring within the home. HomePlug [16][17] has been able to provide customers with Internet service using a modem that connects to the outlets in the rooms of a house. There are several other Power Line Carrier modem manufacturers as well [18][19][20][21][22][24]. This option not only encourages customers to enroll with the utilities demand response program, but then it also provides the customer with the option of using the existing infrastructure to provide Internet access to as many computers in the house that they deem necessary. HomePlug has reported data rates as high 189 Mbps, which is more than sufficient for a simple demand response program. Furthermore, since the devices that are being controlled will be plugged into the outlet which houses the modem the devices energy consumption patterns can be monitored and the state of the device can easily be toggled. All of these features make Power Line Carrier a very appealing technology, however there are several issues that need to be addressed before customers and utilities decide to select this option. The first and most pressing issue is noise. Noise is created by other appliances within the house, which exhibit peculiar nonlinear voltage responses. Figure 2.1 shows the voltage response of four different appliances while in the on position. The small

rectangular voltage responses represent the Power Line Carrier data and the other shapes represent the noise generated by the different appliances. As can be seen for appliances exhibiting nonlinear voltage responses (e.g. Hair Dryer), it can be extremely difficult to determine not only when the Power Line Carrier data signals begin and end, but also determining the amplitude of the signal becomes challenging. In addition to this the channel itself (i.e. wiring within the home) consists of possibly several different types of conductors, which can have a tremendous effect on the signal.

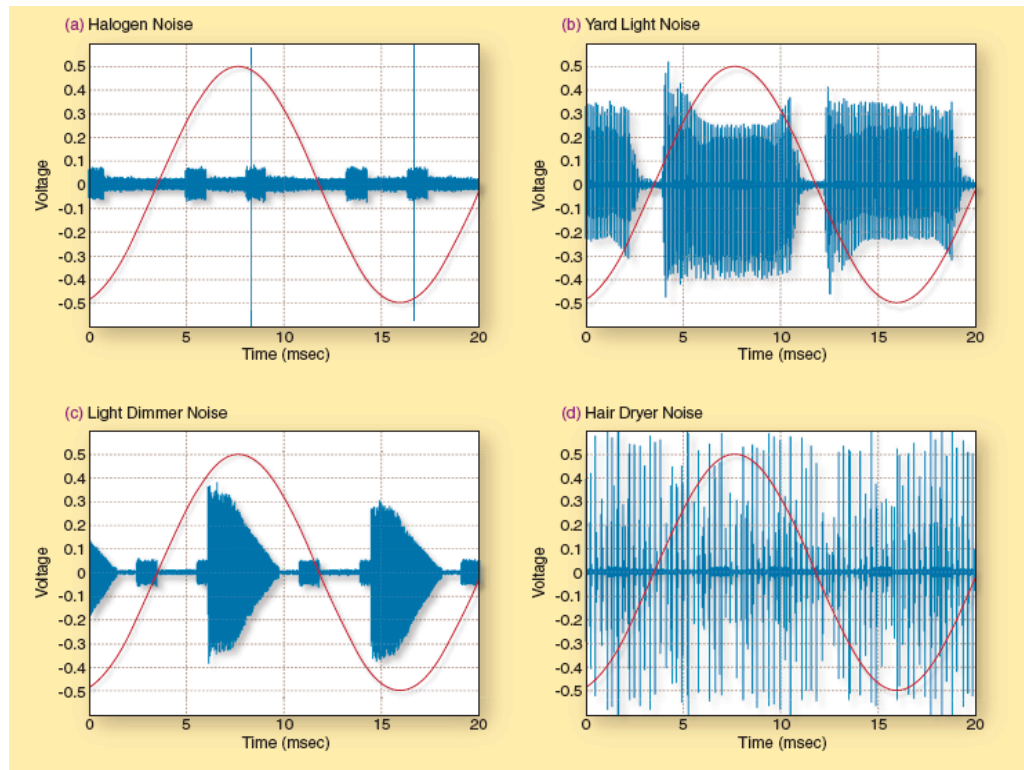


Figure 2. 1 Frequency Response of Appliances

The frequency response of power line carrier channels have been known to experience attenuations as high as 60dB [17]. In order to deal with the noise generated by the appliances within the home Power Line Carrier uses OFDM because most appliances have a cyclical voltage response. This allows the modems connected to the outlets to

differentiate between the Power Line Carrier signal and the noise generated by other appliances operating on the same line. Although OFDM is robust in the face of impulse noise, there are several other components that are needed in order to maximize the probability with which the modem correctly identifies the power line carrier signal. Figure 2.2 shows a block diagram displaying the major physical layer components necessary for the transmission and reception of a signal. For the requirements listed in Table 2.1 the complexity of such a system is not warranted. Furthermore the cost of these devices, which are no less than eighty or ninety dollars U.S., is not justifiable for the simple operations required in Table 2.1. Given the constraints of the coaxial and Ethernet option, as well as the constraints associated with the Power Line Carrier option, the most sensible option would be a wireless solution.

There are a number of home automation wireless protocols that are capable of providing the functionalities outlined in Table 2.1. Table 2.2 provides a list of the most commonly used wireless home automation protocols used today.

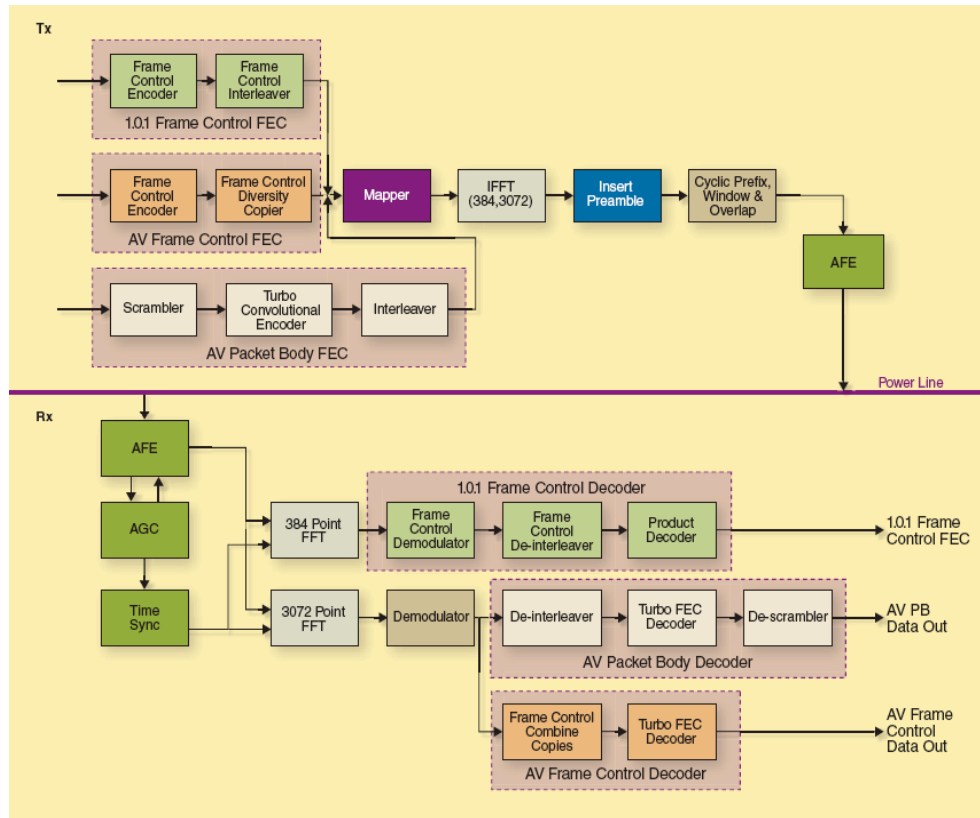


Figure 2. 2 Powerline Carrier Transceiver

Table 2.2 Home Automation Protocols

| | |
|------------------|-----------------|
| Zigbee | Non proprietary |
| EnOcean | Proprietary |
| Low Power 802.11 | Non proprietary |
| Bluetooth | Non proprietary |
| Z-Wave | Proprietary |
| 6LowPan | Non proprietary |

Because of the proprietary issues associated with EnOcean and Z-Wave a number of utilities and vendors have elected to use other protocols instead. Bluetooth and the low power 802.11 protocols are probably the best options, because both of these standards are used to provide many of the services that we use today. However current industry trends have indicated that the preferred option is Zigbee. Zigbee is geared towards low

rate applications (e.g. actuator networks) whereas Bluetooth is used for stream like applications (e.g. connecting a mobile phone to a headset, connecting a mouse to a laptop). The low rate 802.11 protocol is also a viable option, but little interest has been taken in this solution. Furthermore, as will be demonstrated later, the power consumption issues associated with low rate wireless personal area networks can effectively be forgotten if the device uses the appliance as a source of power [9]. Given the trends in the industry and the abundance of literature for Zigbee Home Automation networks, we will use this wireless protocol as the basis for the discussion that follows.

2.1.1.1 Zigbee Home Automation Designs

Several designs have been proposed for a Zigbee Home Automation network [24][25][26][27][28]. In [24] the authors create a wireless network protocol, which is capable of incorporating a plethora of different home appliances, possibly using different protocols, into a single network using the P2P Universal Computing Consortium architecture. They provide a very unique example whereby a video camera can be controlled by a Nintendo Wii controller based on the sensed values reported by a Mica 2 mote. Even though the protocol developed in the paper does coincide closely with the requirements outlined in Table 2.1 the applications developed more closely correspond with event detection (e.g. fire, smoke, intrusion). Whereas the objective of the HAN device, as outlined in Table 2.1 entry “Measurement and Monitoring” is to provide measurements to the user about the state of the appliance. In [25], appliances can be controlled via text message. The system incorporates a centric controller responsible for sending and receiving both GPRS messages and Zigbee messages. Each node is equipped with a Zigbee enabled radio as

well as an actuator that can toggle the state of the devices when necessary. Though this system seems like an ideal candidate the information that is provided to the customer does not comply with the requirements of the HAN device. The only option available to their customers is the toggling of the state of the device. Which does not provide the customer with the information (e.g. how much power is being consumed) necessary in order to make an informed decision. The authors in [28] use sensors and actuators to toggle the state of dimmable lights within a home. First they place light sensing nodes under the source of the light to determine the intensity of the light. Based on the values reported by the sensor nodes the user can make a decision to either increase or decrease the light intensity. Though this design provides the user with meaningful feedback, from which a user can make an informed decision about the light intensity within a certain room, it still does not provide the user with the power consumed and the price to run the appliance. Although these sensors do not provide this functionality, coupling these sensors with a sensor that measures the power being consumed by the lights would provide the necessary HAN device functionality. Several patents have been filed [79][80] for home monitoring systems. The objectives of these systems are to provide constant monitoring of the elderly that live alone. A number of sensors are deployed throughout the home including appliance sensors that monitor the frequency at which certain appliances are used. No mention is made of how the sensors function and what measurements are being taken. Although a household energy consumption monitoring system is implemented presumably using the popular Non Intrusive Load Monitoring algorithms the system does not detail whether the loads can be controlled by the occupants or not. None of the aforementioned systems provide the customer with actual power measurements. Thus, to the best of our knowledge, little research has been done in the way of wireless sensor actuator networks that provide power measurements and the associated

operating costs for running an appliance for specified period of time. There are a number of devices [31][32] that provide the functionality required by a HAN device. Each of these wall socket power meters provides a power measurement as well as the cost to use the appliance. These devices generally lack the wireless functionality, are bulky, and costly [33]. Given the shortcomings of these designs it was our intention to develop a low cost solution that could be used with any appliance. In what follows we provide a very detailed explanation of the current sensor that we used in our experiments.

In the following sections, the details of the Zigbee protocol will be discussed in the context of the application requirements outlined in [15]. A normal relay can be used to toggle the state of an appliance, however this topic is not the subject of this thesis, and will not be discussed. Lastly the sensor responsible for providing power measurements must be integrated with the two aforementioned components. Figures 2.6 and 2.7 give one such implementation.

2.1.2.1 Sensor Design

There are several sensor designs that can be used to monitor the power consumption of different appliances within the home. The power is actually never measured directly, but rather calculated by multiplying the measured root mean square current and voltage values. As a result either the voltage or current has to be known prior to measuring the other value if only one sensing component is to be used. Otherwise two sensors will be needed to measure the voltage and current to compute the power. In this section an introduction and review of the most commonly used current sensors is provided. Using the measurements produced by these sensors as a benchmark, a novel

approach to measuring real power is presented and compared to the results produced by the de facto methods.

The current in a conductor, of any size can be computed in one of three ways. The first method makes use of the Hall effect principal. Whereby a current carrying conductor is placed within a magnetic field, produced by a permanent magnet. A resulting voltage is created across a metal plate attached to the permanent magnet, and is commonly referred to as the Hall voltage. The current in the conductor is proportional to the voltage produced across the metal plate.² The Rogowski coil is another method used to compute the voltage. The Rogowski coil is a toroid of wire that encloses a current carrying conductor and produces a voltage that is proportional to the time rate of change of current through the conductor. Rogowski coils are used to measure the AC current in a conductor.³ Applying Faraday's law to a current carrying conductor provides another means for calculating AC current. Faraday's law states that the voltage, or electromotive force (emf), is equal to the time rate of change of the magnetic flux times the number of turns in the coil.

Each method either requires a coil, or a magnet, to produce a voltage, which is proportional to the current in the conductor. They also require the conductor to be either inside a coil or enclosed by a magnet. One of the major drawbacks of this approach is finding the optimal size coil or magnet to measure the current in conductors of varying size. Conductors of smaller size will produce smaller magnetic fields and as a result, a Hall effect sensor and Rogowski coil would have to be

$$^2 V_H = \frac{IB}{ned}$$

$$^3 V = \frac{-AN\mu_0}{l} \frac{dI}{dt} \quad \text{N-number of turns in coil, } \mu_0 \text{-magnetic constant, } A \text{-area of one of the smaller loops}$$

matched to the corresponding conductor. One-size will not fit all conductors, if either of these two methods are used. Ruling out these two sensor designs would suggest the Faraday current sensor as the optimal choice for devices with different conductor size. The Faraday current sensor has a number of advantages compared to the aforementioned designs. The coil is smaller in size and lighter in weight. It needn't be wrapped around the conductor either thus reducing the time spent on setup. In addition to these properties the Faraday current sensor does not have to be mounted on the conductor with immense precision. It can be mounted on any side of the conductor producing the exact same result, whereas the Hall effect and Rogowski coil require the conductor to be centered in the middle of the sensing apparatus. This adds to setup time as well. Based on this information the ideal candidate for current measurement would be a Faraday current sensor.

Even though each design has its own unique handicap they perform the task that they were designed to do. However this does come at a cost. The voltage-current relationship for the Rogowski coil and Faraday current sensor are linearly dependent on the number of turns in the coil that interacts with the time dependent magnetic field. For appliances with a relatively small current draw the number of windings in the coil should be substantially large enough to actually capture the effect of the magnetic field due to the time varying current in the conductor. Otherwise the calculation of the current based on the voltage values collected by the microcontroller will suffer from significant measurement errors. To address all of these concerns when selecting a current sensor, requires an in depth knowledge of the appliance and how it operates. Appliances that draw current on the order of microamperes can be difficult to measure if proper design is not taken into consideration. Thus emphasizing the fact that a one size fits all approach using one of the aforementioned sensors will only

work in some cases. These problems can be resolved, and have been addressed in the [34][35]. The authors present a new sensor design that accurately reproduces the current in a conductor of varying size and current by using a piezoelectric bimorph. The current measurements in [34] closely coincide with the actual values and the authors subsequent follow up work in [35] suggest that the sensor can be reduced in size to capture even smaller current values. An additional feature of this sensor design is its ability to provide the device that it is performing the measurements on, with the energy necessary to power the microcontroller and wireless radio. With such an approach, the energy being harvested by the sensor, while in generator mode, provides all of the necessary energy to power both the microcontroller and the wireless radio. After investigating the claims and reproducing the results in [34] the optimal choice was apparent and the choice was made to use this design in a prototype design of a wireless current sensor. In the next section a detailed discussion about the piezoelectric effect and the pertinent equations for reproducing the current in a conductor are discussed. Unlike the expressions for the other sensors the relationship between the current and the voltage is related through a second order differential equation. As a result a working knowledge of electromechanics is essential in understanding this relationship in order to understand the computational requirements of the microcontroller or ESI for multiple devices.

2.1.2 Piezoelectricity

Piezoelectric material has been, and currently still is being used, in a variety of sensing applications [36]. The piezoelectric effect consists of electromechanical coupling. A piezoelectric crystal is a crystal capable of producing an electric potential (voltage) as a result of some external stimuli other than electricity. If the crystal is deformed in any

way (e.g. compressed, stretched, sheer forces) a resulting electric potential is created proportional to the force(s) applied to it. This is most commonly referred to as the direct effect. The crystal will also deform if an electric potential is applied to it and this is referred to the converse effect. As with the direct effect the forces exerted by the crystal on the objects in direct contact with it is directly proportional to the electric potential applied to it. The direct effect is the operational mode that is used and therefore the analysis that will be performed below will not provide an in depth explanation of the equations describing this effect.

There are a number of ceramic materials that exhibit piezoelectric behavior, however the most commonly used and preferable ceramic is Lead-Zirconate-Titanate (PZT) for reasons that will become clear later. It is easiest to study ceramic materials and their corresponding behavior by analyzing the properties of the fundamental building blocks of the material. These fundamental building blocks are most easily understood if they are viewed as crystals uniform in: length, width, and height. Although it is not necessary to use this approach to analyze the element of the material under consideration it greatly simplifies the analysis and development of the expressions needed to relate certain forces to electric potentials and vice versa.

Using the coordinate system in Figure 2.3 as an example of a crystal of uniform length in all three dimensions, a series of mathematical relationships can be established between the stress and strain experienced in the material and the resulting electric potential, and vice versa.

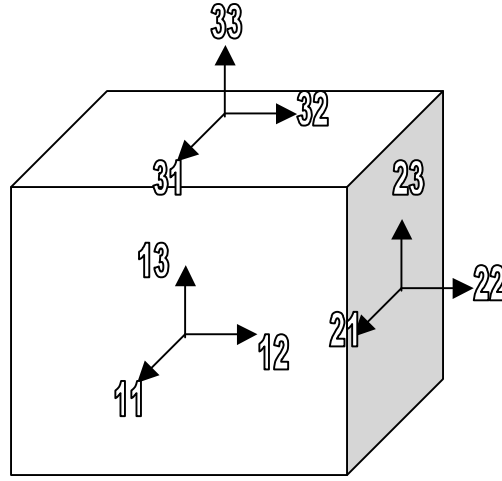


Figure 2. 3 Unit Cube of Piezo Material

Either a stress is applied to piezoelectric material and a resulting electric displacement is measured or an electric field is applied to the material and the displacement of the material is observed. These two different scenarios are referred to as the direct and converse piezoelectric effects respectively. The direct and converse piezoelectric effects are modeled mathematically by the following equations.

$$S = sT \quad (2.1)$$

$$D = dT \quad (2.2)$$

$$D = \epsilon E \quad (2.3)$$

$$S = dE \quad (2.4)$$

2.1.2.1 Direct Piezoelectric Effect

The direct piezoelectric effect is explained mathematically by a linear relationship between the applied stress, denoted by T (N), and the resulting strain S (m/m) as indicated in equation (2.1). When a stress is applied to an object the crystals parallel to the axis upon which the stress is being applied are elongated to a certain length Figure 2.4. The strain is therefore defined as the total change in length, or elongation, relative

to the unperturbed length of material. The linear relationship between the stress and strain only holds up until a critical stress value, which is dependent on the material being used, after which the strain will be either constant (hardening material) for all values of applied stress or will increase without bound for all values of stress applied (softening material). The constant s (m^2/N) is the mechanical compliance and is the inverse of Young's modulus.

In addition to the strain experienced by the material and the crystals therein, because of the electrophysical properties of the material an electric potential will also be produced by a charge flow created by this movement. Each crystal consists of a collection of electric dipoles. The dipoles in the crystal have the potential to align themselves in such a way as to produce an electric charge flow as illustrated in the right most figure in Figure 2.4. Once the material has been stretched the electric dipoles begin to point in the same direction (dipole rotation) thus creating an electric field that can be measured by a pair of electrodes connected to the material. This effect is defined as the ratio of the total charge produced to the total area occupied by the electrodes connected to the material, which is called the electric displacement D (C/m^2). The relationship between the applied stress and the electric displacement is linear too over a certain range of values as indicated by equation (2.2), but beyond a critical value the relationship becomes nonlinear and the material will become saturated with charged particles thus producing no additional electric displacement. The constant d (C/N) represents the piezoelectric strain coefficient and its properties are dependent on the material being used.

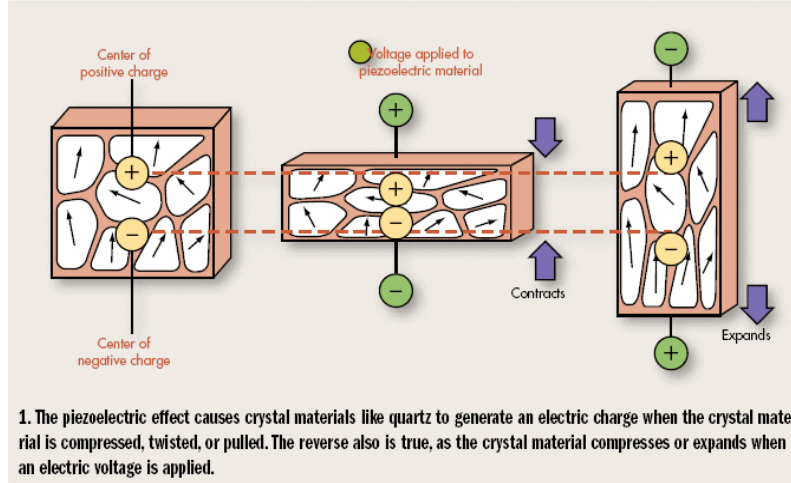


Figure 2. 4 Piezoelectric Effect [37]

2.1.2.2 Converse Effect

Piezoelectric material also exhibits a reciprocal effect whereby an applied electric potential will create an electric field E (V/m) within the material equal to the applied electric potential divided by the distance between the electrodes connected to the material. As a result the electric dipoles in the material will be attracted to the electric field produced in the material and dipole rotation will commence. This rotation of the dipoles will create an electric displacement linearly dependent on the electric field and subsequently the applied electric potential. This relationship is captured in equation (2.3) and the constant ϵ (F/m) is the dielectric permittivity. Although a linear relationship exists between the electric field and electric displacement, beyond a critical electric field value the material becomes saturated with electric dipoles pointing in the same direction. Thus creating the same electric displacement for all subsequent electric field values.

In addition to the electric displacement of the charged particles the material also

experiences a stress as described above and explained in equation (2.4). As with the other linear relationships, the strain experienced in the material will only be linear up to a certain point after which the material will begin to experience a softening and therefore continuous elongation Figure 2.4.

2.1.2.3 Direct, Converse Piezoelectric, and Constitutive Equations

Equations (2.1)-(2.4) can be combined into matrix notation and described as a relationship between the two mechanical variables, stress and strain, and two electrical variables, the electric field and electric displacement.

$$\begin{Bmatrix} S \\ D \end{Bmatrix} = \begin{bmatrix} s & d \\ d & \varepsilon \end{bmatrix} \begin{Bmatrix} T \\ E \end{Bmatrix} \quad (2.5)$$

This matrix forms the basis for the constitutive relationships between the stress and strain as well as the electric displacement and electric field. Using the cube in Figure 2.3, and the associated coordinate system, a general result can be created relating the mechanical and electrical properties of any structure made of piezoelectric material.

2.1.2.4 Electric field and Electric Displacement

Assuming that an electric field can be applied to any side of the face, or that an electric field can be applied to multiple faces at the same time the relationship between the electric displacement of charged particles and the applied electric field becomes the following.

$$\begin{aligned}
D_1 &= \varepsilon_{11}^T E_1 + \varepsilon_{12}^T E_2 + \varepsilon_{13}^T E_3 \\
D_2 &= \varepsilon_{21}^T E_1 + \varepsilon_{22}^T E_2 + \varepsilon_{23}^T E_3 \\
D_3 &= \varepsilon_{31}^T E_1 + \varepsilon_{32}^T E_2 + \varepsilon_{33}^T E_3
\end{aligned} \quad (2.6)$$

$$\vec{D} = \begin{matrix} \vec{\varepsilon}_1^T \\ \vec{\varepsilon}_2^T \\ \vec{\varepsilon}_3^T \end{matrix} \vec{E} \quad (2.7)$$

Each equation relates the electric displacement of charged particles on a face of the cube to the electric fields through all faces of the cube. The superscript signifies that a constant normal or sheer strain is being applied to the corresponding face indicated by the subscripts. Albeit a very plausible scenario, in our application the electric field will only be present in one direction, which will reduce the number of equations needed. As a result the expressions (2.6) and (2.7) can be rewritten as follows.

$$D_m = \varepsilon_{mn}^T E_n \quad (2.8)$$

2.1.2.5 Stress and Strain

The stress and strain relationships for all three faces can be computed in a similar manner. However the shear and normal stresses must be taken into consideration. There are six faces on the cube resulting in exactly three stress components normal to the faces of the cube and six sheer stress components along the sides of the cube. Applying either a normal or sheer stress to the cube will produce a nine different strain values. Thus a total of 81 different stress and strain values need to be calculated in order to relate the stress to the strain.

$$S_{ij} = \Psi_{ijkl}^E T_{kl} \quad (2.9)$$

The coefficient Ψ_{ijkl}^E relates the stress applied to the $(k,l)^{\text{th}}$ face of the cube to the strain experienced by the cube on the $(i,j)^{\text{th}}$ face when the material is subject to a constant electric field in any direction.

2.1.2.6 Strain and Electric Field

As was shown in the one-dimensional case, when an electric field is applied to piezoelectric material the charged particles will experience a strain in the same plane that the electric field vector is in. Although this is generally true for most materials depending on the size and configuration of the material, some materials will experience nine different strains. Three of these components will be normal and the other six will be sheer components. In general if an electric field is applied to one of the faces then the material will undergo a strain with these nine different components.

$$S_{ij} = \Delta_{ijn} E_n \quad (2.10)$$

There are 9 different coefficients that correspond with an electric field applied to the face of the cube.

2.1.2.7 Stress and Electric Displacement

A similar relationship exists between the stress applied to a cube and the electric displacement that the charged particles experience. However in the case of a mechanical stress applied to a cube there will only be an electric displacement in one of three directions for all nine components of stress. As an example, if a sheer stress is applied in the 21 plane, then the charged particles in the cube will experience an electric displacement in the three directions parallel to the principal axes. The same

will hold for all nine stress components applied to the cube.

$$D_m = \Delta_{mkl} T_{kl} \quad (2.11)$$

The relationships developed in equations (2.8)-(2.11) can be combined and summarized in the following compact set of equations.

$$\begin{aligned} S_{ij} &= \Psi_{ijkl}^E T_{kl} + \Delta_{ijn} E_n \\ D_m &= \Delta_{mkl} T_{kl} + \epsilon_{mn}^T E_n \end{aligned} \quad (2.12)$$

These two equations describe all of the possible interactions between an applied stress and electric field and the resulting strain and electric displacement. The entire set of equations is described by: 81 mechanical compliance constants Ψ_{ijkl}^E , 27 piezoelectric strain coefficient values Δ , and 9 dielectric permittivity constants ϵ_{mn}^T . Equation (2.12) is referred to in literature as the constitutive equations for linear piezoelectric material. This representation of the constitutive equations becomes difficult to deal with especially if a thorough investigation of the material of interest is warranted. Depending on the application the material is being used for, an analysis of the resulting strain and electric displacement of the charged particles in the material will require at least 108 equations if an electric field is not present. Using a set of symmetrical tensor relationships the computational complexity of equation (2.12) can be greatly reduced. Assuming that the stress and strain tensors are equal, when applied along the same coordinate axis, the (i,j)th stress and strain components are equal to the (j,i)th stress and strain components. Using this result the stress and strain components can be redefined as follows.

$$\begin{aligned}
S_1 &= S_{11} & T_1 &= T_{11} \\
S_2 &= S_{22} & T_2 &= T_{22} \\
S_3 &= S_{33} & T_3 &= T_{33} \\
S_4 &= S_{23} + S_{32} & T_4 &= T_{23} = T_{32} \\
S_5 &= S_{13} + S_{31} & T_5 &= T_{13} = T_{31} \\
S_6 &= S_{12} + S_{21} & T_6 &= T_{12} = T_{21}
\end{aligned} \tag{2.13}$$

Using this notation equation (2.12) can be rewritten in matrix notation with 36 independent mechanical compliance constants, 18 piezoelectric strain coefficients, and 9 dielectric permittivity values thus making the analysis of piezoelectric material behavior more tractable.

$$\begin{Bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{Bmatrix} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} \\ s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} \\ s_{31} & s_{32} & s_{33} & s_{34} & s_{35} & s_{36} \\ s_{41} & s_{42} & s_{43} & s_{44} & s_{45} & s_{46} \\ s_{51} & s_{52} & s_{53} & s_{54} & s_{55} & s_{56} \\ s_{61} & s_{62} & s_{63} & s_{64} & s_{65} & s_{66} \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{Bmatrix} + \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \\ d_{41} & d_{42} & d_{43} \\ d_{51} & d_{52} & d_{53} \\ d_{61} & d_{62} & d_{63} \end{bmatrix} \begin{Bmatrix} E_1 \\ E_2 \\ E_3 \end{Bmatrix} \tag{2.14}$$

$$\begin{Bmatrix} D_1 \\ D_2 \\ D_3 \end{Bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{Bmatrix} + \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{32} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{bmatrix} \begin{Bmatrix} E_1 \\ E_2 \\ E_3 \end{Bmatrix} \tag{2.15}$$

Although not explicitly indicated in expression (2.14) it is assumed that the mechanical stress constants are the constants associated with a material under constant exposure to an electrical field s_{ij}^E . In expression (2.15) a similar relationship exists

where the dielectric permittivity constants ϵ_{kl}^T represent the dielectric permittivity when a constant stress is being applied to the material.

$$\vec{D} = d\vec{T} + \epsilon^T \vec{E} \quad (2.16)$$

$$\vec{S} = s^E \vec{T} + d' \vec{E} \quad (2.17)$$

Adding three dimensions to the analysis results in an electric displacement vector and strain vector of size 6×1 which are both functions of the stress and electric field applied to the material as before, however the piezoelectric strain coefficient, mechanical compliance, and dielectric permittivity are all matrices now of size 3×6 , 6×6 , and 3×3 respectively. Thus 63 coefficients must be determined in order to show the relationships between stress, strain, electric field, and electric displacement. Most of the values are provided in data sheets for applications in which the material will be subject to a variety of different stresses and electric fields. However in our application the material will only be subject to a stress along one of the principle coordinate axis and the coefficients associated with the other principal axes are eliminated from the analysis. Although this reduces the number of coefficients necessary to form a relationship between the electric potential and the applied stress, it will be shown that the elimination of these coefficients will only have a negligible effect on the measured electric potential.

2.1.2.8 Measured voltage as a function of tip deflection

In our sensor design we used a (PZT) piezoelectric bimorph fashioned after the design proposed in [34]. The dimensions of the bimorph as well as the permanent magnets correspond exactly with those in the paper. The subsequent analysis of the electric potential measured across the electrodes connected to the piezoelectric bimorph as a

function of the tip deflection with the magnets on the end can be reduced to a simple mechanics problem. The piezoelectric bimorph can be viewed as a beam cantilevered to a fixed point in space with a length, l , greater than both the width, w , and height, h , and a height smaller than either of the other two dimensions as shown in Figure 2.5.

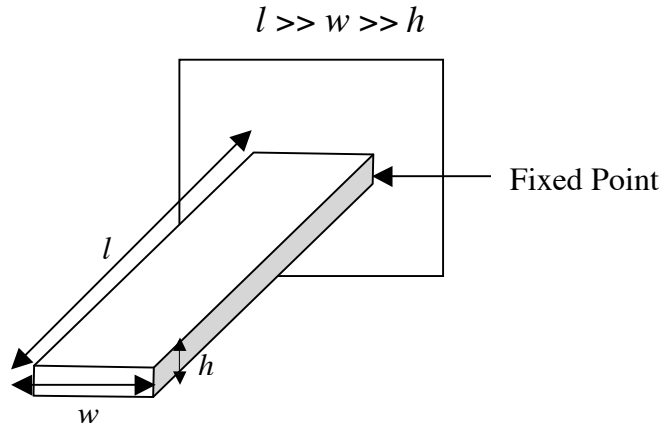


Figure 2. 5 Cantilevered Beam

To simplify the analysis of the stress and electric field relationship it is assumed that a normal force applied to the tip of the beam will only result in a stress along the length of the beam. This reduces the number of equations to one stress equation along the length of the beam and one electric displacement equation normal to the plane that the length of the beam is in.

$$\begin{aligned} S_1 &= s_{11}^E T_1 + d_{31} E_3 \\ D_3 &= d_{31} T_1 + \epsilon_3^T E_3 \end{aligned} \quad (2.18)$$

Using these equations it has been shown that the electric potential generated by the freely flowing electric charges in the piezoelectric material can be measured [38]. In this paper the authors attached a mass to the free end of a piezoelectric beam and

generated a force normal to the mass by moving the cantilevered end of the bimorph up and down in an oscillatory motion. The oscillation of the beam creates an oscillatory electric displacement of the charged particles in the piezoelectric material. Furthermore the authors reduce the mechanical portion of the piezoelectric generator to an electric circuit, from which a relationship between the strain vector of the mechanical circuit and the voltage of the electric circuit can be formed. This method of analysis was used to derive a relationship between the force applied to the free end of the piezoelectric generator and the electric potential it produces [35]. The following equations are produced as a result of performing Kirchoff's Voltage and Current Laws to the mechanical and electrical portions of the circuit respectively.

$$\ddot{S} + 2\xi_m \omega_n \dot{S} + \omega_n^2 S = \frac{k_{sp} a_1 d_{31}}{m t_p} V + \frac{F_{in}}{k_2 m} \quad (2.19)$$

$$\dot{V} = \frac{a_3 c_p d_{31} t_p}{a_2 \epsilon} \dot{S} \quad (2.20)$$

Applying the Laplace transform to both equations yields two expressions the first of which can be substituted into the second equation thus providing a linear relationship between the force applied to piezoelectric generator and the electric potential produced.

$$V_s = \frac{a_3 c_p d_{31} t_p}{a_2 \epsilon} S_s \quad (2.21)$$

$$s^2 S_s + 2\xi_m \omega_n s S_s + \omega_n^2 S_s - \frac{k_{sp} a_1 d_{31}}{m t_p} V_s = \frac{F_{in}}{k_2 m} \quad (2.22)$$

The subscript, s , has been added to both the electric potential and strain variables to indicate that a transformation has been made from the time to frequency domain.

Upon substituting equation 2.21 into equation 2.22, the electric potential generated by

the piezoelectric generator (bimorph) can be expressed as a linear function of the force applied to the tip [35]. The result is reproduced here, for convenience.

$$V_{out} = F_{in} \left[\frac{a_3 c_p d_{31} t_p}{a_2 k_2 m \varepsilon \left[(j\omega)^2 + 2\xi_m \omega_n (j\omega) + \omega_n^2 \left(1 - \frac{c_p d_{31}^2}{\varepsilon} \right) \right]} \right] \quad (2.12)$$

$$V_{out} = F_{in} K(d_{31}) \quad (2.13)$$

$$F_{in} = I\beta \quad (2.14)$$

$$V_{out} = I\beta K(d_{31}) \quad (2.15)$$

2.2 Current Sensor design

The design of a vibration-based bimorph to generate an electric potential as a function of force was used in [34] to measure the current in a conductor. Instead of moving the cantilevered end of the device up and down to create the oscillation of the mass on the free end, the authors use the interaction between the magnetic field produced by the current in the conductor and a magnet attached to the free end to generate the oscillations. Although the oscillations that the mass experiences are similar, the way in which the oscillations are generated are different and have a different effect on the piezoelectric material. In [38] the oscillations are generated using a vibrometer to simulate how the bimorph might generate energy for a wireless sensor located in a remote environment (e.g. habitat monitoring, battlefield, roadways). As a result when the bimorph is displaced, the cantilevered part and the free end with the attached mass are displaced and this is not captured by the constitutive equations in [38]. However the expressions developed in [34] neglect the movement of the cantilevered end, and assume that it is fixed. Which for the purposes of measuring the current in a conductor, with a stationary-cantilevered end, is sufficient.

In [34] the mass used in [38] is replaced by two rare earth permanent magnets. The current in the conductor creates a magnetic field directly above the centerline of the conductor causing the magnet to move up and down much like the mass in [38]. Although the conductor consists of two wires carrying current in opposite directions the results still apply to a single conductor with one current carrying wire. The only difference in the expressions for the magnetic field located above the conductor is the distance between the centers of the two conductors. So the results can be generalized to any kind of configuration once the geometry of the conductor is understood.

The force that the magnet experiences due to the magnetic field generated by the conductor can be computed by using the following relationships assuming that the magnet is treated as a dipole.

$$\vec{F} = \nabla(\vec{m} \cdot \vec{B}) \quad (2.16)$$

The magnetic dipole moment is denoted by \vec{m} , the del operator is denoted by ∇ , and the magnetic field by \vec{B} . The dipole moment can be replaced with the magnetization of the material and, because the magnet is a permanent magnet the magnetization can be replaced by the residual flux density. Depending on the orientation of the magnet with respect to the current carrying conductor, the force on the magnet can be greater or smaller. In our experiments we consider the magnet to be located directly above the conductor. The force equation simplifies to the following for this particular orientation.

$$F_y = B_r \int \frac{d(H_y)}{dy} dV [39] \quad (2.17)$$

$$\frac{d(H_y)}{dy} = -\frac{i}{\pi} \frac{2dy}{(y^2 + d^2)^2} \quad (2.18)$$

The volume and the residual magnetic flux density \vec{B}_r are both constant. Therefore the force and current have a linear relationship as demonstrated in equation (2.14).

2.2.1 Current as an Instrument for Power

Given the relationships developed above, the electric potential generated by the piezoelectric generator, can be digitized with an analog-to-digital converter (ADC) to reconstruct the current in the cord. Figures 2.6 and 2.7 display: the current sensor, a circuit used to step down the voltage within an acceptable range for the ADC included in the MICAz microprocessor, and the MICAz wireless radio.

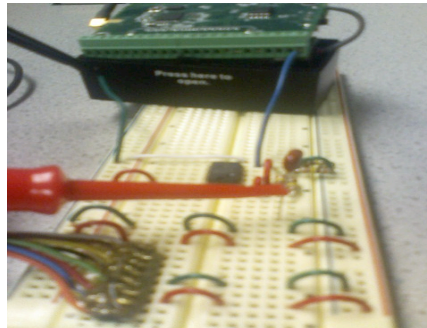


Figure 2. 6 Current Sensor coupled with MICAz

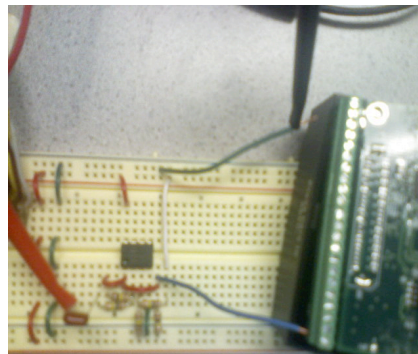


Figure 2. 7 Current Sensor coupled with MICAz

The sensor design that we elected to use only provides the current waveform, but the objective is to compute the real power. It would be possible to use another sensor that measures the voltage (electric potential) of the appliance, but that would make the design too bulky. Plus as will be shown, not only would it be more challenging to synchronize the voltage and current sensor outputs, the power measurements based on the product of these two values grossly overestimates the actual power being consumed [40]. Instead we use another method to calculate the power consumed by an appliance, using only the current measurements. This is based on the fact that a nonlinear relationship exists between the power being consumed and the current being drawn by the appliances. A regression of the real power on the current will allow us to make an estimate that is relatively close to the actual power. The actual power consumed by an appliance is called the apparent power.

$$S = P + jQ$$

The apparent power is a linear combination of the real power, measured in Watts (W), and the reactive power, measured in volts-amperes reactive (VARs).

$$\begin{aligned} P &= I_{RMS} V_{RMS} = I_{RMS}^2 R \text{ for a purely resistive load} \\ Q &= I_{RMS} V_{RMS} = I_{RMS}^2 |X| \text{ for a purely reactive load} \end{aligned}$$

In actuality both the real and reactive power is being measured however utilities are only allowed to charge residential customers for the real power that they consume unless of course they are operating large devices that require an extraordinary amount of reactive power. The utility is only able to charge customers for the amount of real power being consumed, depending on the appliance type, even though the appliance consumes a certain amount of reactive power. If the real power is significantly greater than the reactive power consumed by the appliance as it cycles through the different stages, then the reactive power is considered to be negligible. Even though the

relationship between the real power and the current is non-linear an approximate linear relationship taking the form outlined below can be developed.

$$\begin{aligned}
P_{AVG_i} &= \beta_0 + \beta_1 I_{RMS_i} + \beta_2 I_{RMS_i}^2 + \varepsilon_i \quad (2.17) \\
\vec{P}_{AVG} &= \begin{bmatrix} \vec{1} & \vec{I}_{RMS} & \vec{I}_{RMS}^2 \end{bmatrix} \vec{\beta} + \vec{\varepsilon} \\
\vec{P}_{AVG} &= \begin{bmatrix} \vec{1} & \vec{x}_1 & \vec{x}_2 \end{bmatrix} \vec{\beta} + \vec{\varepsilon} \\
\vec{P}_{AVG} &= X\vec{\beta} + \vec{\varepsilon}
\end{aligned}$$

The number of rows in the matrix X , correspond with the number of samples taken. The error term ($\vec{\varepsilon}$) contains other variables that might not have been included in the explanatory variable matrix (X), or it could contain other statistical disturbances that cannot accurately be modeled using a linear regression. The authors in [40] showed that, for a series of 212 Hot Water pumps, a linear regression of the real power on a constant, the current, and the square of the current would provide estimates for the coefficients (β_i) that could be used to compute subsequent real power values. The percentage difference of the actual and computed values was no larger than 2.5% for a twenty-four hour period with a sampling interval of 1 minute. Furthermore this technique is used for most appliances within the home, which do not have total harmonic distortions that vary frequently. This would include: HVAC systems, dishwasher, washers and dryers, refrigerators, and pool pumps. Appliances that have switching loads, draw current at irregular frequencies thus creating frequency components, which are multiples of the standard 60 Hz system operating frequency. These appliances are usually computers, office equipment, or any device that has a power supply. Power supplies normally have rectifiers, which create non-linear components in the power signatures.

In order to use the current as a proxy for the power, we used a similar setup to that

outlined in [40]. As our appliance of choice we selected the refrigerator. Using a Watts Up Pro power meter, we logged the current and power measurements over a twenty-four hour time period with a sample period of two seconds. The coefficients are shown in Table 2.3. This eliminates the need for additional sensors to compute the voltage. Furthermore as shown [40] the proxy power measurements are far more accurate than the power measurements calculated from multiplying the current and voltage. This can be seen in Figures 2.8 - 2.10. Figure 2.8 displays plots of the: measured power(actual power), the product of the voltage and the current, and the current as a proxy power estimate for a period of fifty minutes. However the coefficients were calculated using data from an entire 24-hour period. One sample was collected each second. The power calculation based on the product of the current and voltage overestimates the measured power, and is also greater than the estimated power. The time period corresponding with Figure 2.9 and Figure 2.10 is an hour and half and three quarters of an hour respectively. We tested multiple refrigerators to determine if there was a significant difference in the coefficients calculated for the different samples Table 2.3. Our original hypothesis was that the coefficients should not vary significantly from one sample (i.e. refrigerator) to another. The reason being that, the underlying mechanics of a refrigerator are the same regardless of the manufacturer. The results suggest that there is little to no difference between the three regressions. Therefore the linear regression provides a reasonable estimate of the power produced by an appliance and the margin of error between the actual value and the predicted value is small enough to warrant using this method as a means to estimate the power being consumed. Looking at the plots, it is apparent that for the older refrigerators(7+ years old for sample 1 and 5+ years old for sample 2) the difference between the estimates and the actual values is greater. Whereas for a new refrigerator(e.g. 2-3 years old - sample 3), the difference between the estimates and the actual value is a lot smaller.

Table 2.3 Linear Regression Coefficients

| Sample 1 | Sample 2 | Sample 3 |
|---------------------------|----------------------|----------------------|
| $\beta_0 = -2.1657526968$ | $\beta_0 = -2.0007$ | $\beta_0 = -0.2608$ |
| $\beta_1 = 128.785866102$ | $\beta_1 = 126.1921$ | $\beta_1 = 124.4052$ |
| $\beta_2 = -7.0434760355$ | $\beta_2 = -5.2471$ | $\beta_2 = -10.4764$ |
| $R^2 = 0.9529$ | $R^2 = 0.9711$ | $R^2 = 0.9883$ |
| Time: 16:51 | Time: 23:16 | Time: 22:00 |
| % difference | % difference | % difference |

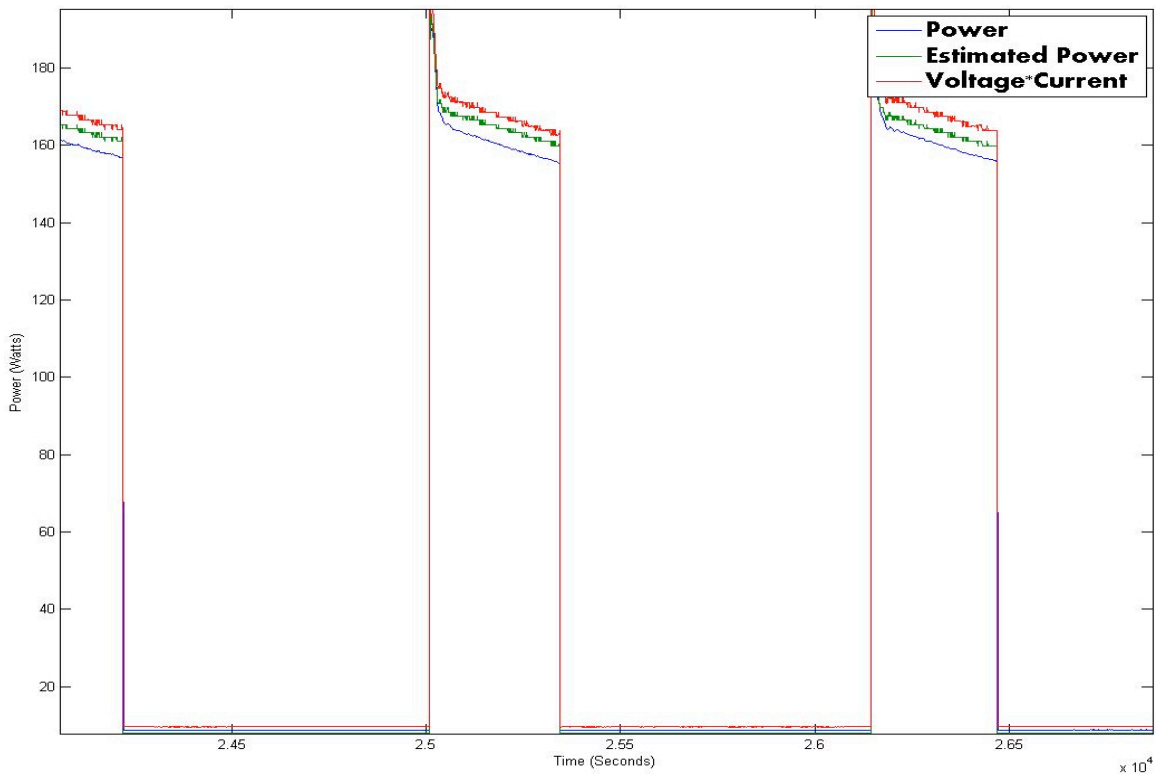


Figure 2. 8 Power vs Time for Sample 2

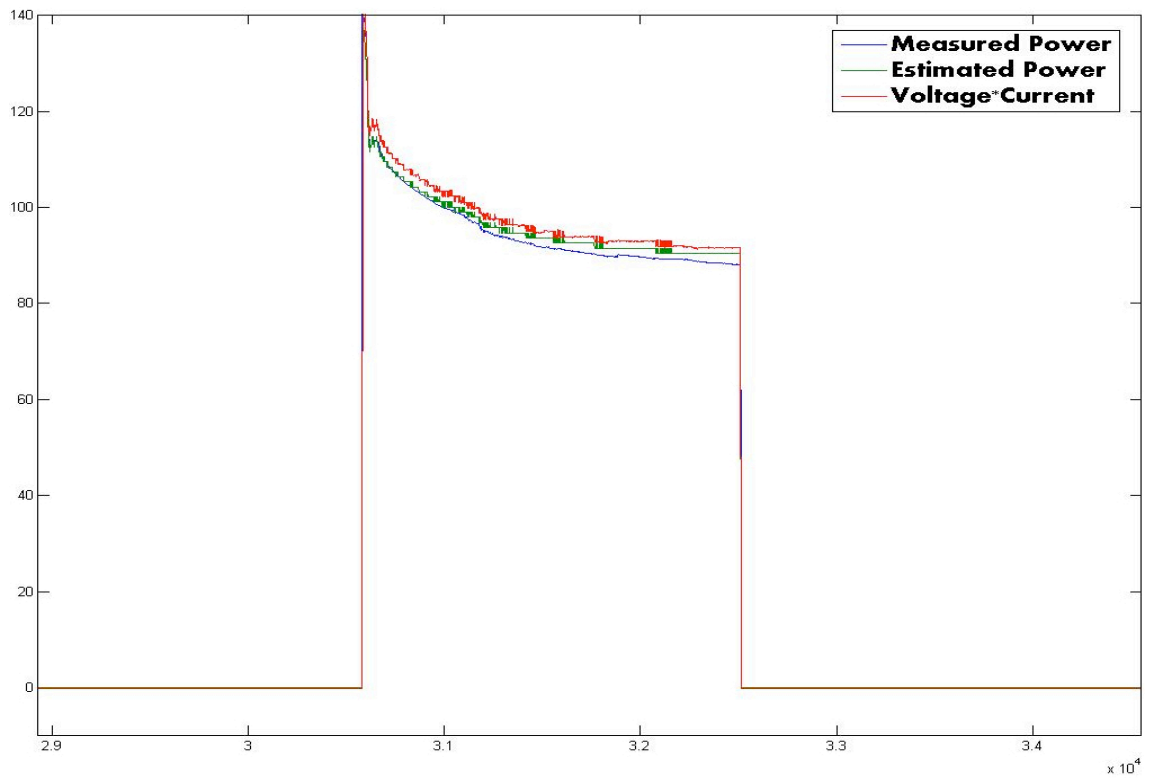


Figure 2. 9 Power vs Time for Sample 3

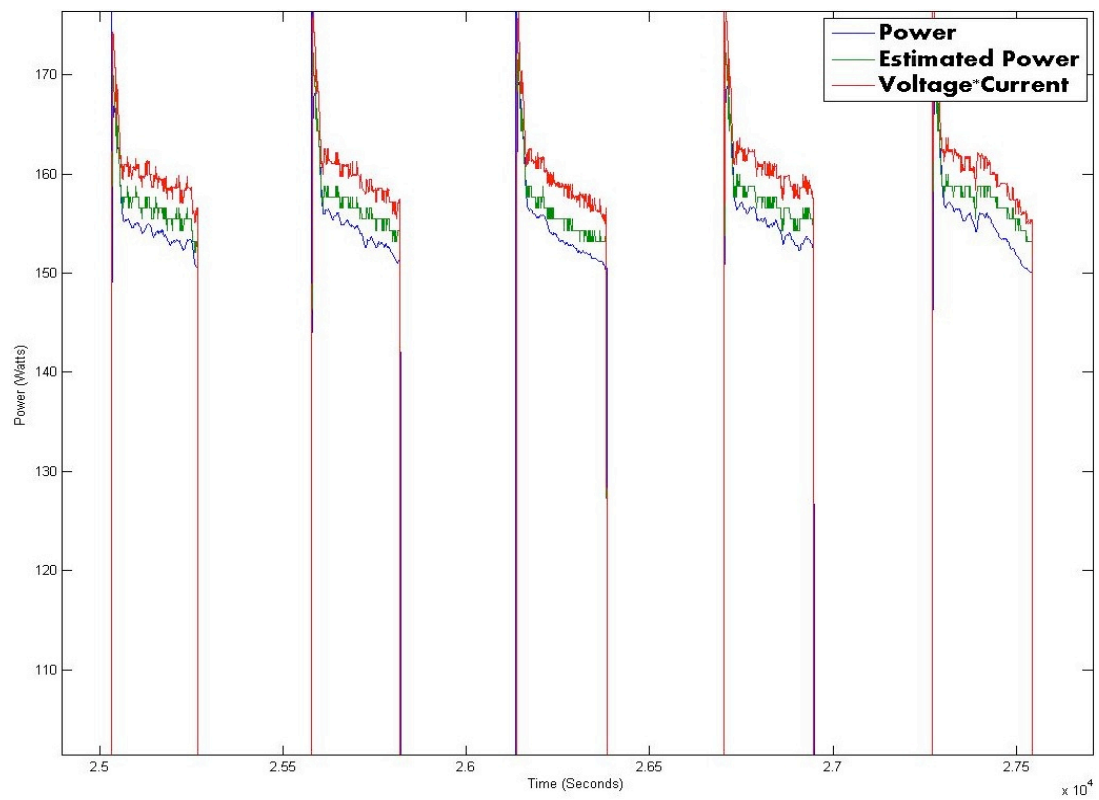


Figure 2. 10 Power vs Time for Sample 1

Looking forward this entire design can be integrated with an ADC included in any standard microprocessor, along with a wireless radio. The samples of the raw voltage signal produced by the sensor can be used to compute the RMS voltage. After which the RMS current can be computed. Using the RMS current values, and the linear regression coefficients, which will presumably be stored in memory on the node attached to the appliance⁴ [41], the power can be computed. The RMS current values will have to be computed using the samples generated by the sensor. A sufficient number of samples will have to be generated per second in order to compute the RMS values for the current. For each raw output voltage value generated by the sensor a corresponding current value will be computed on the node using the relationship developed in (2.15).

$$V_i \rightarrow I_i$$

$$I_{RMS} = \sqrt{\frac{\sum_{i=1}^n I_i^2}{n}} \quad (2.15)$$

$$I_{RMS} \rightarrow P_{AVG}$$

The number of samples is represented by n . Once the RMS values have been computed the samples can be deleted from memory and the corresponding power values can be computed. This will be essential for low power devices using the Zigbee protocol.

⁴ We assume that the manufacturer will provide the coefficients because power and current ratings must be measured for each cycle an appliance goes through to ensure that the appliance meets the new regulatory standards [41]

2.3 Zigbee and 802.15.4 Overview

Zigbee was originally created for projects that have lifetimes of months or perhaps even years. It is a routing protocol that is used in conjunction with the IEEE 802.15.4 specification for low rate wireless personal area network (LWPAN) radios to optimize the lifetime of the devices being used. Dedicated sleep and wake cycles are created based on the topology of the network, and the hierarchical structure imposed by the network engineers based on the objectives of the sensing applications. As an example, for a habitat-monitoring project, a two-tiered network will be critical in order to collect the data necessary for scientists to infer meaningful information about the animals grazing in a certain area. However as will be seen in the following chapter, a multi-tiered approach leads to single points of failure.

The promise of being able to deploy and leave a network unattended for several years is a very attractive feature. This coupled with the fact that the messages are relatively simple to use, makes it an even more attractive option to utilities looking for an easy to use demand response solution for their customers.

2.4 Overall System Design

In the system that we consider, the meter serves as a PAN (Personal Area Network) coordinator, or a full-function device (FFD), for each appliance, or reduced-function device (RFD) in the network [11]. In addition to the meter's ability to coordinate appliances within the Home Area Network (HAN), the meter will also provide a communication link to the Neighborhood Area Network (NAN) access point using a 802.15.4 radio. Given the fact that 802.15.4 radios can successfully transmit packets a

distance of 50 meters—nearly half the length of a football field—the meters can form either a mesh or star network with other meters in the neighborhood. Within an average size residential dwelling, the distance between the appliance and the meter will be at most 30 meters. Thus appliances can transmit packets using a lower power setting. Appliance registration can be accomplished, if a customer has enrolled in a direct load control program with their electric utility. This would in turn give the utility the ability to toggle the state of an appliance within the home when a system wide blackout might be imminent. This being said the utility should keep customer data private and, under no circumstance should a utility release customer data to an unauthorized entity. As will be shown in the next section if the sensors are used inappropriately, very private and meaningful customer information can easily be revealed [42]. However if the following communication and control requirements are met for HAN devices as outlined in [15]: confidentiality, non-repudiation, availability, and integrity can be guaranteed for all communications. Most of these requirements, if not all, are met or exceeded by the Zigbee standard as outlined [43].

CHAPTER 3

Neighborhood Area Network

3.1 Neighborhood Area Network Introduction

In this section we introduce the next component of the demand response system, which is the Neighborhood Area Network. This consists of the meters attached to the houses, along with the access point, which forwards customer data to the utilities local office. Although a standard Neighborhood Area Network(NAN) definition does not exist yet, a few members of the UCAIug have identified components, which they deem necessary for successful communication between the residence and the utility. The network would consist of an AMI meter equipped with an 802.15.4 radio using the Zigbee Pro networking stack along with the Smart Energy(SE) profile. The SE profile defines the standard behavior of secure, easy-to-use, Home Area Network (HAN) devices [24]. The radio is responsible for appliance-to-meter communication and meter-to-access point communication.

We will begin with the reactive routing protocol that serves as the basis for the Zigbee Pro networking protocol.

3.1.1 AODV

Zigbee is a hybrid routing protocol that incorporates elements of both AODV and Cluster Tree routing. Cluster tree routing is only used on Reduced Functional Devices (RFDs), which are usually battery operated or resource constrained devices. AODV was created for use by Full Functional Devices (FFDs), which are less resource constrained than RFDs because they are usually tethered to a power supply. In order to establish a personal area network, first a PAN coordinator must be elected. After a

PAN coordinator has been selected other nodes looking to join the PAN must send a request to join and will be permitted to join based on availability of coordinator resources (e.g. memory and energy). As outlined in [44] cluster tree routing is an optimal choice for applications with very low duty cycles. Furthermore most of the routes used by nodes in a PAN using cluster-tree routing are not optimal in hop count resulting in an uneven traffic distribution. This is due to the fact that, nodes closer to the root of the tree have to route more traffic to the PAN coordinator [44]. In addition to this networks using the multi-tiered architecture suffer from single points of failure. As show in Figure 1.1, if a cluster head were to fail then all of the nodes within that PAN are rendered useless. Given these reasons, the choice to use a pure AODV routing protocol will ensure that such network segmentations will not occur.

AODV (Ad hoc On-Demand Distance Vector) routing is one of several reactive routing protocols. Originally created for mobile ad hoc networks (MANETS), AODV proved to be a resilient routing protocol when the topology of a network was unknown, thus making it very attractive for randomly deployed ad hoc wireless sensor networks. The protocol can be described in short, as one in which a route is established between a source and destination only when the source has packets to send to a destination. Otherwise, the node is idle. Essential to the establishment and the maintenance of routes between any two source and destination pairs is the concept of sequence numbers. A sequence number is guarantees that the path being used between the source and destination pair consists of nodes, which have the most up to date routes to the destination. Destination sequence numbers are essential for networks with a high degree of mobility to ensure that the paths are recent enough for communication to commence. In a static network sequence numbers are beneficial because if a path becomes unavailable due to an obstruction or some other unforeseen

issue, then another path can be established. This eliminates the single point of failure problem that cluster-tree networks are prone to experience.

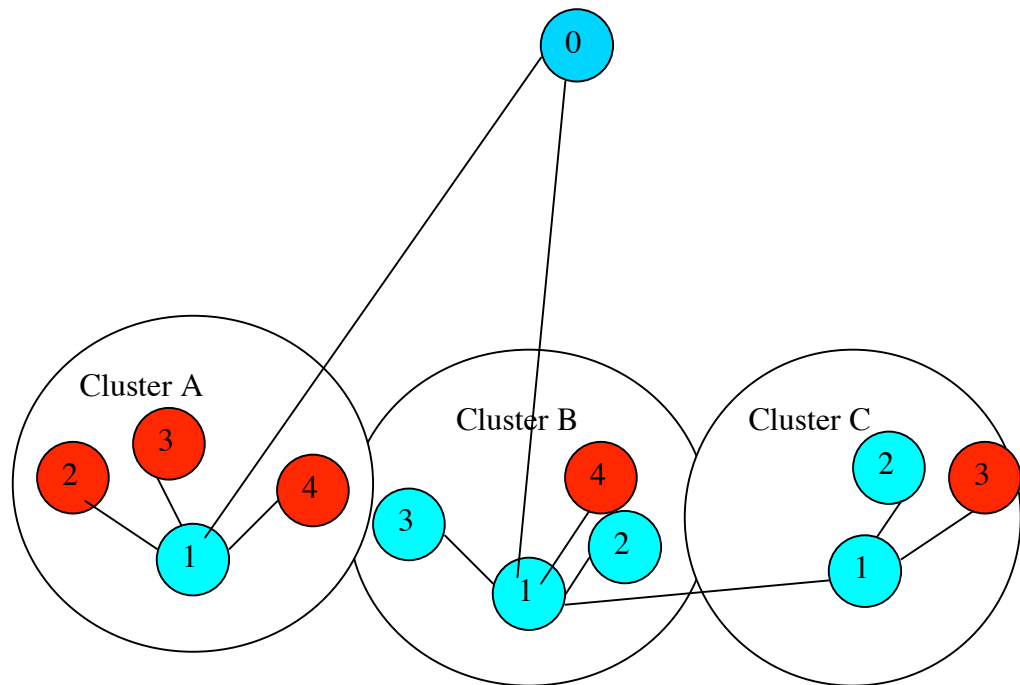


Figure 3. 1 Cluster Tree Network

Table 3. 1 Example Network Legend

| |
|-----------------|
| PAN Coordinator |
| FFD |
| RFD |

Control messages are used to help a source node ascertain a path to the destination. These messages include route request (RREQ), route reply (RREP, route error (RRER), and hello messages (HELLO)[45]. Each of these control messages are sent

from the Transport layer, to the Internet layer using UDP along with the regular IP header. The control messages are also used by intermediate nodes, to learn about paths to certain destinations within the network. Each node is responsible for maintaining route table entries for all destinations that it has established a path with either for its own communication, or for other nodes downstream from it. In addition to this, each node must maintain a list of all nodes that have used it to forward packets to a destination in the network, and are one hop away and downstream from it. This list is called a precursor list and is essential for forwarding along route reply messages to the nodes requesting a route to the destination. The next section contains a brief overview of the AODV protocol.

3.1.1.1 Route Discovery

A node will issue a RREQ only when that node determines that it needs to establish a route to a destination within a network. This occurs in one of two situations. The first is if a route to the destination does not exist, and the second is when a previously established route to the destination has been lost. For extremely mobile networks the second scenario occurs frequently resulting in a network, which is constantly flooded with RREQ packets. However for a static network a RREQ should only be issued enough times to establish a route to the destination once, because the nodes in the network are not mobile. Before the packet is sent, if a node does not have a route to the destination as described in scenario one above, then the unknown destination sequence number flag bit must be set to either zero or one. RFC 3561 does not specify whether a one or zero corresponds with the flag being raised, and is therefore left to the discretion of the network engineer. An originator sequence number is necessary for the dissemination of the corresponding RREP message that will be sent back to the

source from the destination. A RREQ ID is also included in the packet to ensure that replayed or multiple RREQ messages received by intermediate and destination nodes are dropped once they are received. For each subsequent RREQ issued the RREQ ID is incremented by one. To enable bidirectional communication, as will be necessary for an application whereby smart meters report energy usage values and also receive pricing information, the source node should set the gratuitous RREP flag.

| <i>Type</i> 8 bits | <i>J</i> 1 bit | <i>R</i> 1 bit | <i>G</i> 1 bit | <i>D</i> 1 bit | <i>U</i> 1 bit | <i>Reserved</i> 11 bits | <i>Hop Count</i> 8 bits |
|---|-------------------|-------------------|-------------------|-------------------|-------------------|----------------------------|----------------------------|
| <i>RREQ ID</i> 32 bits | | | | | | | |
| <i>Destination IP Address</i> 32 bits | | | | | | | |
| <i>Destination Sequence Number</i> 32 bits | | | | | | | |
| <i>Originator IP Address</i> 32 bits | | | | | | | |
| <i>Originator Sequence Number</i> 32 bits | | | | | | | |

Figure 3. 2 Route Request Message Format

The hop count is also maintained in the RREQ message as it propagates through the network to the destination.

As the RREQ propagates throughout the network, intermediate nodes are responsible for checking that the packet contains a valid sequence number, as this provides loop free routing [45]. A node receiving a RREQ, from another node one hop away from it without a valid sequence number, will create a route table entry for the source and destination sequence pair. If the node has received a RREQ within the last PATH_DISCOVERY_TIME, as outlined in [45] with the same RREQ ID, the node will drop the RREQ. A node forwards a RREQ by first incrementing the hop count by

one then looks for a reverse route to the node with the Originator IP Address (i.e. source node) listed in the RREQ packet. The reverse route is necessary for the RREP that will be sent from the destination back to the source through this node. If a reverse route to the source node does not exist then one is created using the Originator Sequence Number in the RREQ. Once the route has been created, the Originator Sequence Number is copied into the Destination Sequence Number field for the entry corresponding with the Originator IP Address. Then the valid sequence number field is set to true, the next hop in the routing table is updated to be the node from which the RREQ has been received, and the Hop Count in the RREQ packet is copied into the hop count field of the entry corresponding with the Originator IP Address.

An intermediate node will only continue to forward the RREQ if the Destination Sequence Number in the RREQ is larger than the Destination Sequence Number contained in its routing table. The destination sequence number corresponding with the entry in the routing table for the Destination IP Address will be set equal to the larger of the two Destination Sequence Numbers. In this case it would be the one contained in the RREQ. However if the node does contain a sequence number which is greater than the one contained in the RREQ it will generate a RREP packet. In order to generate a RREP the node copies the Destination IP Address and the Originator Sequence Number from the RREQ into the RREP. Then the RREP is unicast back to the next hop along the reverse path that it received the RREQ from. The node from which the RREQ was received is referred to as a precursor. Each node maintains a precursor list of all nodes, one hop away, from which it has received a RREQ in order to establish the reverse route. The reverse route is used to propagate the RREP back to the node issuing the RREQ. As the RREP traverses the network back to the node originating the RREQ each node in between will increment the Hop Count so that the

total hop count that the source node receives is equal to the distance between itself and the destination.

As mentioned before a gratuitous RREP will be generated if the gratuitous flag in the RREQ packet is raised. The gratuitous RREP will have the same message format as that of a RREP message, but the information contained therein will be different. In the hop count field it should contain the hop count corresponding to the route table entry for the originating node. The Destination IP Address field will be the IP address of the node, which originated the RREQ, and the sequence number field should contain the Originator Sequence Number contained in the RREQ. Finally the Originator IP Address field should be the IP address of the node for which the RREQ was destined. Then the gratuitous RREP is sent to the Destination IP Address listed in the RREQ, as if the destination node had sent a RREQ to the node with the Originator IP Address.

| <i>Type</i> 8 bits | <i>R</i> 1 bit | <i>A</i> 1 bit | <i>Reserved</i> 9 bits | <i>Prefix Size</i> 5 bits | <i>Hop Count</i> 8 bits |
|---|-------------------|-------------------|---------------------------|------------------------------|----------------------------|
| <i>Destination IP address</i> 32 bits | | | | | |
| <i>Destination Sequence Number</i> 32 bits | | | | | |
| <i>Originator IP address</i> 32 bits | | | | | |
| <i>Lifetime</i> 32 bits | | | | | |

Figure 3. 3 Route Reply Message Format

If an intermediate node is the recipient of a RREP message it will search for a route to the previous hop from which it received the RREP. If one does not exist then it will create one. If one does exist, then the recipient node will check the Destination Sequence Number in the RREP with the Destination Sequence Number in its route

table entry corresponding with the Destination IP Address in the RREP message. The sequence number in the table will be updated if the sequence number in the table is invalid, the sequence number in the RREP is greater than the one contained in the table, the sequence numbers are the same but the route for that entry has been marked as invalid in the passed, or the sequence numbers are the same but the Hop Count field in the RREP message is less than the Hop Count for that table entry. As soon as the RREP has been transmitted, the precursor list that corresponds with an entry in the node's table is updated by including the node, which is the next hop towards the node that originated the RREQ. Lastly the precursor list for the node that is the next hop closest to the destination is updated to include the next hop closest to the source. Once the forward and reverse routes have been created the node can use the two routes to begin forwarding data packets between the source and destination pair.

3.1.1.5 Path Maintenance

In order to maintain connectivity during an active communication session nodes can issue HELLO messages to update important routing table information. A node accomplishes this by checking whether it has sent a broadcast message (e.g. RREQ or HELLO message) within a predefined period of time designated by the HELLO_INTERVAL. If the node determines that a message has not been sent within this time period then it will broadcast a RREP message with the Time To Live (TTL) field set equal to one, the Destination IP Address to its own IP address, the Destination Sequence Number equal to its own latest sequence number and a Hop Count of zero. By definition this is a HELLO message [45]. For a node trying to determine the connectivity to its nearest neighbors, the node should leave its receiver on to listen for packets. If a route to a destination using these nearest neighbors had been established

in the past, and a HELLO message or data packets have not been received within the past DELETE_PERIOD, then the link to its neighbor is assumed to be broken. If instead a node does receive a HELLO message the recipient node checks to see if a route to the sending node exists. If one does not then one is created otherwise the entry in the table for the node issuing the HELLO message is updated to include the latest destination sequence number, and the Lifetime for the route that the link is being used for is increased. After the routing tables have been updated the node will continue to send, or begin sending, data packets between the source and destination pair.

3.1.1.5 Local Connectivity

Each intermediate node that forwards packets to and from different destinations has to maintain local connectivity with the nodes that belong to its precursor list and next hop lists, as well as any neighbors that it receives periodic HELLO messages from. This can be accomplished by using a link layer notification system or by passive acknowledgment. If a message is not received within NEXT_HOP_WAIT seconds, as specified by the implementer, then the link is assumed to be broken and a RERR message must be issued.

| <i>Type</i> 8 bits | <i>R</i> 1 bit | <i>Reserved</i> 15 bits | <i>Hop Count</i> 8 bits |
|---|-------------------|----------------------------|----------------------------|
| <i>Unreachable Destination IP address (1)</i> 32 bits | | | |
| <i>Unreachable Destination Sequence Number</i> 32 bits | | | |
| <i>Additional Unreachable Destination IP Addresses (if needed)</i> 32 bits | | | |
| <i>Additional Unreachable Destination Sequence Numbers (if needed)</i> 32 bits | | | |

Figure 3. 4 Route Error Message Format

If the link is assumed broken then the existing routes that depended on the link are marked as invalid, the destinations and neighbors affected by the break in the link are listed, and a RERR message is sent to the affected neighbors. A RERR message is initiated if any of the three following things happen. If a node detects that the next hop, downstream from it, has been broken while transmitting data then a RERR message will be sent to the nodes in the precursor list affected by the break in this link. If the node receives a packet for a node that it does not have an active route for and will not be repairing then a RERR message will be issued. Lastly if a node receives a RERR message from neighboring nodes for one or more active routes, then a RERR message will be issued.

The neighbors of this node that should receive the RERR message are those that belong to the precursor list of one or more of the unreachable destinations in the RERR message. The RERR message is generally broadcast to all neighbors to notify all other nodes of the broken link that was once used to communicate with the unreachable nodes. Before the REER is transmitted, for a node generating the RERR, the destination sequence number for each one of the unreachable destinations in the route table entry is incremented. For a node that is forwarding a RERR the sequence number for the unreachable destination in the route table entry is updated to the sequence number in the RERR message. The entry corresponding with each of the unreachable destinations is then marked invalid and the Lifetime field of the entry should be updated.

3.1.1.5 Local Repair

If a link break occurs during an active communication session, the node that is upstream from the break should repair the link only if the destination of the data packets are only MAX_REPAIR_TTL hops away. In order to repair a broken link a node must first increment the sequence number for the route table entry corresponding with the destination and then a RREQ is broadcast. The RREQ is sent to the destination in order to reestablish a connection. To ensure that the RREQ is given enough time to reach the destination, or an intermediate node, with an up to date route to the destination. The time to live (TTL) value is set to the maximum of, the last known hop count to the destination or the hop count to the originator of the undeliverable data packets, plus a constant value of 2 representing the local TTL to be added to the packet [45]. After the RREQ is sent, the node will wait an entire discovery period for a RREP. If a RREP is not received within this period of time a RERR message is issued as described above.

However if a node does receive: one or more RREPs, other control messages, or a combination of the two, within the discovery period, the node compares the hop count in the message for the destination with the hop count for the destination in the route table entry. If the hop count in the received message is greater than the hop count for the destination in the route table, the recipient of the control message sends out a RERR message with the N bit flag set. The route table entry is then updated for that destination and then a RREP message is sent to the node originating the undeliverable data packets. When a node performs a local repair, it buffers the messages for the destination and once it has been repaired it sends them. Although this can result in longer paths to the destination, the packets that have been buffered on the repairing

node will not be lost as they would be if the originating node decided to repair the link itself. If the originating node decides to repair the link itself, then the discovery process for that node will recommence.

Although several destinations might be affected by the break in a link, the node upstream from the broken link will only repair the link towards the destination for which the received data packets are bound. All other routes using the link are marked invalid for the time being, but the repairing node will designate each route that has been lost in its route table as being locally repairable. The routes will only be repaired though as packets for the unreachable destinations arrive at the repairing node. As the network becomes congested, the node may begin to repair the routes to the other unreachable destinations even before a data packet for a particular destination arrives. As a result any data packets that arrive for a previously unreachable destination, will not be delayed. If however the route to the destination is no longer being used, then time will have been wasted repairing a link that is not being used.

3.1.2 AODVJr & Zigbee

The Zigbee Specification [46] uses a modified version of the AODV protocol outlined above called AODVJr [47]. This protocol does not include: sequence numbers, Gratuitous RREPs, Hop Count, Hello Messages, RERRs, or precursor lists. The changes to the protocol were made in part to reduce the number of control messages used when establishing routes to and from a destination. However as outlined in both [46][47] the use of destination sequence numbers has completely been eliminated. This has the unfortunate consequence of forcing each node requesting a route to a destination to issue a RREQ and then wait for a corresponding RREP from the

destination. Intermediate nodes are not allowed to respond to RREQs, leading to higher delays during the route discovery period [48]. Sequence numbers are essential for maintaining loop free paths between any two nodes in the network. Proof II in the Appendix explains why the destination node may only respond when destination sequence numbers are not used with AODVjr. The destination sequence numbers, along with the other fields included in AODV control messages [47], are not included to reduce control overhead.

A path in [46] is defined in terms of an ordered set of devices(nodes), and the length of this path is defined as the cardinality of the ordered set.

$$P \equiv \{D_1, D_2, \dots, D_L\} \text{ with } L = \#P$$

L is the cardinality of the set P and represents the length of the path. A link is a sub-path of length 2 and the cost of the entire path is the sum of the costs associated with sending a packet down each one of the links.

$$\begin{aligned} C\{P\} &= \sum_{i=1}^{L-1} C\{(D_i, D_{i+1})\} \\ LC_{(i,i+1)} &= C\{[D_i, D_{i+1}]\} \\ LC_{(i,i+1)} &= \begin{cases} 7, \\ \min\left(7, \text{round}\left(\frac{1}{p_i^4}\right)\right) \end{cases} \end{aligned}$$

The probability with which a packet is successfully sent between two consecutive nodes is denoted by p_i . The value for p_i is determined on a case by case basis, but usually is determined to be a function of the Link Quality Indicator as defined in [49]. The link costs are used in place of destination sequence numbers in order to maintain up to date routes to destinations in the network. As a result for every control message

sent (e.g. RREQ or RREP), the cost associated with each link must be calculated as the packet traverses each link to its destination. For large networks consisting of several thousand nodes and one sink, the delay experienced by nodes further from the sink will increase and the throughput of the nodes further from the sink begins to decrease as well [50]. Thus using sequence numbers would allow nodes further from the sink node, especially along the edges of the network, to establish a route quickly and efficiently. This is because intermediate nodes, with an up to date route, will send a RREP back to the originator of the RREQ indicating that a route to the sink exists. The delay and the number of dropped packets decrease exponentially if this approach is used [45].

Another field that is not included in Zigbee control messages is the hop count. It has been replaced by the path cost. There are two path cost fields that are maintained in the node's Route Discovery Table. The first is the Forward Cost, which is a measure of the path cost associated with a RREQ as it travels from a source node to the destination. As the RREQ traverses the nodes in the network towards the destination each node will update the path cost field in the RREQ and each node will store that value in the Forward Cost field in its Route Discovery Table. The second path cost field in the Route Discovery Table is the Residual Cost and is the path cost associated with the RREP message as it propagates through the network from the destination towards the source node. In the Zigbee protocol, nodes will only update these two Route Discovery Table entries if the Forward and Residual Cost values in the RREQ and RREP messages are less than the values in their Route Discovery Table entries. As the nodes in the network continue to update their Forward Cost entries and the corresponding Sender Addresses (i.e. nodes that they receive RREQs from), if an intermediate node mistakenly issues a RREP to the node that it overheard the RREQ

from a routing loop will be created. This is outlined in the Appendix and proven in Proof II. Thus each intermediate node must forward each RREQ that it overhears to the destination to maintain loop freedom. This has the unfortunate consequence of increasing the overhead and the associated discovery time.

RERRs are not used in the specification either and link repair is declared to be outside of the scope of the specification. This is due in part to the fact that a precursor list is not maintained. In a Zigbee network each node will issue a Link Status message, periodically, indicating whether or not a link has been broken. This message is broadcast to all nodes in the network. It is then left up to the node receiving the Link Status message to determine which destinations in their route table have become unreachable. The precursor lists in AODV are used not only to setup a reverse path to the originator of the data packets, but also to notify the nodes in the precursor list of any links, between a source and destination pair, that might have become broken during an active session. These messages are generally unicast to each node in the precursor list unless there are multiple precursors in which case it will be multicast to all members in the list. Zigbee nodes learn about link breakages by overhearing a message that is broadcast by its neighboring nodes. Each neighbor of the node issuing the link status message will look at the list of nodes included in the message for its address. For each address in the list there exists a status indicator based on the cost of the link between the recipient and the sender. The status indicator is based on the incoming and outgoing cost associated with each neighboring pair, and is used by the recipient to update the link status associated with the sending node. Although this provides each node in the list with link status information about its neighbor's neighbors, it still does not inform the nodes in the list about the destinations that have become unreachable. The only information that is provided is link cost information.

Which is not enough information for a node to surmise that the neighboring node, which sent the Link Status message, will deem a link broken. The sending node therefore does not provide enough information to its neighbors to determine which destinations have become unreachable due to link failures. The aforementioned shortcomings are concerns that must be considered for time sensitive applications such as demand response systems. Electric power demand response systems must provide close to real time information about customer energy usage to ensure that the electric power system continues to operate within stable operating region. Timing is essential in order to maintain tolerable voltage and current values, correct amounts of real and reactive power, and a reasonable power factor for customers that are a part of the distribution system. This particular application is the focus of this thesis, and in the sections that follow our suggestions and modifications to the Zigbee and AODV routing protocols are presented.

3.2 Smart Meters & Demand Response Systems

The electric power system has three main components: generation, transmission, and distribution. Both generation and transmission are instrumental in providing electricity to the end users who are connected to the distribution system. Figure 3.4 displays a power plant, used for generation, high voltage transmission lines, and the distribution system, which consists of all connections between the power substation and the transformer drum. Figure 3.5 is an image of the future electric power system. All three components exist in this figure as well, however a layer of communication has been added to each component enabling reliable communication between the system operators and the three components of the system.

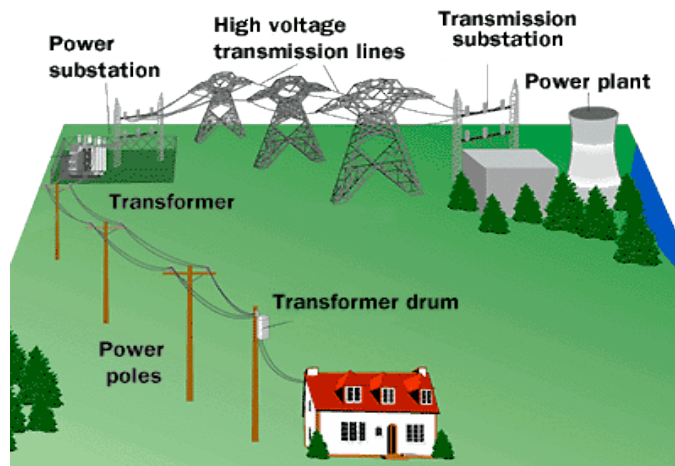


Figure 3. 5 [51] Legacy Power System/Grid

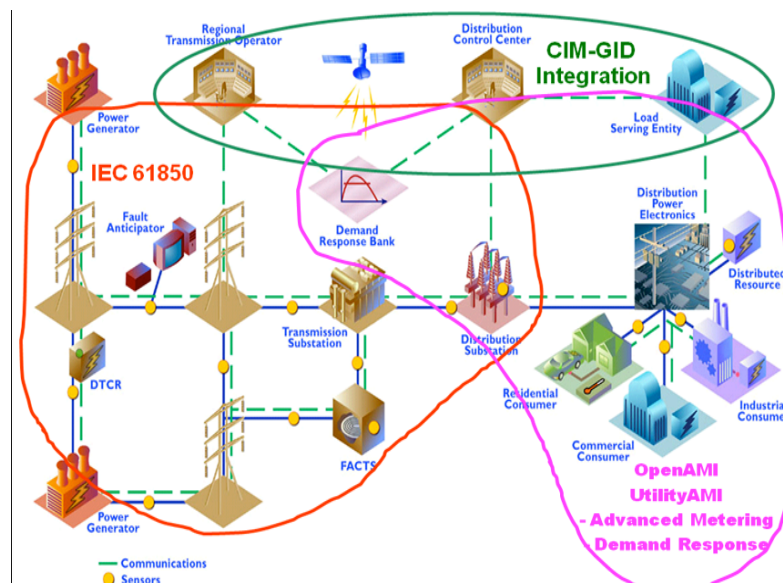


Figure 3. 6 Potential Smart Grid Illustration

The generators are no longer just an energy producing plant isolated from the rest of the grid, but now have the ability to communicate with one another using the IEC 61850 communication protocol, and also with system operators who are responsible for guaranteeing that the load demanded is equal to the available generating capacity.

Using the communication network provided by the transmission system the Regional Transmission Operator (RTO) is able to communicate with the generating units and also monitor the transmission lines delivering power to transmission and subsequently distribution substations. Also, as shown in Figure 3.5, the RTO and Distribution Control Center (DCC) share a wireless link enabling the two entities to schedule the appropriate time periods when certain generators will be needed in order to meet the demand. Lastly the distribution system is equipped with advanced (smart) metering devices, which are integrated with the demand response system. This level of interconnectedness virtually allows the customer's meter, in the distribution, to communicate with the generators on the opposite end of the system. Although the meter would not have the ability to talk directly to the generators providing the electricity, the information collected at the customer premise will obviously influence the behavior of the generators. As the demand for energy decreases the number of generators that are forecasted to be online will decrease, and those that are operational will have their outputs reduced. This has the effect of reducing the amount of energy being consumed and consequently reduces the carbon footprint of the generators offered into the market. Communication between both endpoints of the system is necessary in order to regulate, and potentially reduce the unnecessary expulsion of noxious gasses into the atmosphere. Demand Response is defined as "Changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized." by the Federal Energy Regulatory Commission [52]. Though the definition focuses on the benefit to the customer, system wide benefits are realized if customers respond according to pricing signals. Notably the generation companies will more accurately be able to forecast the number

of generators and the corresponding output levels for the remainder of the day and for the following day.

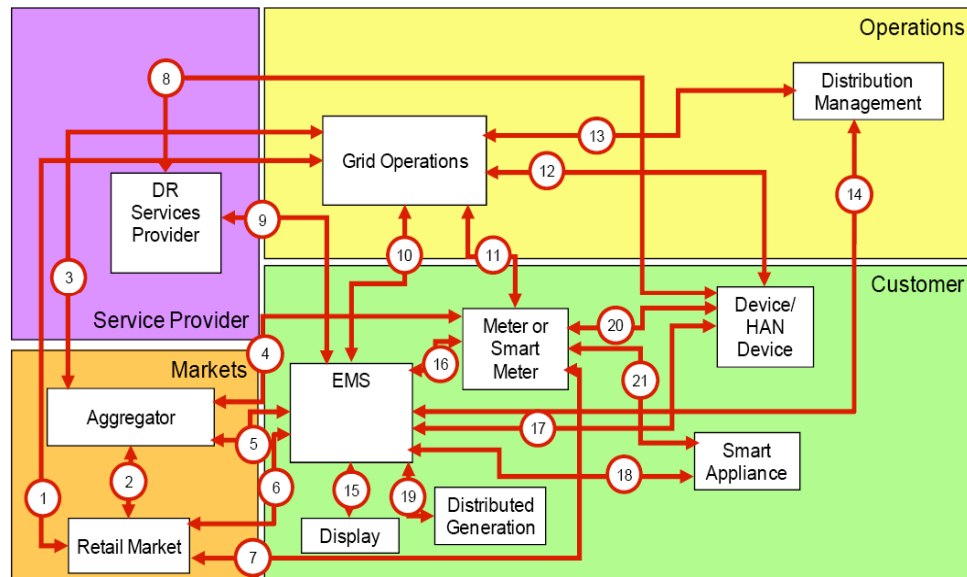


Figure 3. 7 Advanced Metering Infrastructure Interfaces [53]

The RTO can use demand response information to keep: voltage, current, and power values, both real and reactive, from going above or below a certain level on the transmission lines by performing a similar forecasting methodology as used by the generators. The procedure for performing contingency analysis, and the procedure for predicting the occurrence and location of a current fault, can be enhanced by incorporating the information provided by a demand response system. This is perhaps one of the most important aspects of a demand response system, because network faults can leave customers without power for hours.

Demand response systems, in the distribution network, usually consist of a collection of smart meters [43][55] tasked with successfully routing demand response messages

to an access point located in the distribution system that is connected to the utilities Meter Data Management System residing on a server or group of servers. Each meter is connected to either a residential, commercial, industrial, and possibly in certain circumstances municipal customers. Thus creating a large network of smart meters. Some utilities have several thousand meters communicating with just one access point as described above. Several analytical, simulative, and experimental works have shown that the delays and throughput begin to increase and decrease respectively as the size of the network increases [50]. In the seminal paper [50] the authors showed that the per node throughput in a network, is a function of the inverse of the number of nodes in the network. Thereby, asymptotically decreasing the throughput and increasing the delays without bound. Furthermore the authors in [57] show through simulation that the throughput, although different from the results shown in [57], is still a function of the inverse of the number of nodes in the network. The results provided in these papers suggest that the throughput in the networks being considered for the smart metering infrastructure will begin to decrease as the number of nodes added to the network begins to increase. This can be mitigated by using a hierarchical approach, whereby each node belonging to a subset of nodes, $\{s_i\}$ $i=1,\dots,n$, are responsible for collecting information from another subset of nodes, $\{s'_j\}$ $j=1,\dots,m$ where $m \gg n$. However, this results in a single point of failure. If one of the nodes belonging to set $\{s_i\}$ fails, then all other nodes that use s_i to relay its packets will not be able to communicate other nodes in the network. This suggests that a flat topology would be more advantageous. The results shown in [58] prove that a flat topology is preferable when connectivity is an issue in large-scale metropolitan networks.

3.2.1 Smart Meters

A smart meter can loosely be defined as an electric meter capable of providing: current, voltage, and real and reactive power measurements remotely to a utility using any two-way communication mechanism. Currently most utilities that are deploying smart meters are choosing to use a wireless option to provide the communication necessary to collect this information. Most early adopters of smart meters are using the IEEE 802.15.4 wireless Medium Access Control and Physical layer standard, along with the Zigbee Pro specification to provide the two-way communication requirement [46]. For a network of 500,000 smart meters, with a sampling period of every 15 minutes, approximately 400×10^6 bytes of data will be generated per year per meter [59]. Each node sends data hop-by-hop using neighboring nodes as routers to route their data to a super node (sink). The super node is connected to a MDMS (Meter Data Management System), which resides on the utilities servers, via Ethernet. The sheer size of the network, and the amount of information, albeit miniscule relative to the aggregate, will contribute to congestion and decrease throughput. As Figure 3.19 shows, a network of 100 nodes, inclusive of the sink, with a sampling period of 900 seconds, has an average network delay of 300 milliseconds. During regular operating conditions, when the forecasted load is less than or equal to the supply of energy being produced, it is safe to assume that the average network delay will be 300 milliseconds. However when the sampling rate increases, the network will begin to experience contingencies. An increased sampling period might be warranted, when there is an unexpected increase in the demand for electricity (e.g. a hot day). In order to guarantee that these objectives are met, the current ad hoc protocols being used must be modified to increase the likelihood that each node's packet is successfully delivered to the sink

during the allotted time.

3.2.1.1 Time Intolerant Wireless Sensor Networks

There are several ways in which to combat the congestive problems of wireless sensor networks. The first and most commonly used technique is scheduling. The most obvious and trivial way in which to schedule the node transmissions would be to select nodes at random in the network and allow enough time for the node to successfully send its packet to the sink. It has been shown however, that even if a random source and destination pair are chosen, and the nodes in the network have been programmed to send their packets according to a predefined schedule based on their locations, the lower bound for the data rate between the source and destination pair is

$$C = \frac{kW}{(1 + \Delta)^2 \sqrt{n \log n}} [57].$$

Though it has been shown [60] that the capacity of a link between multiple sources and just one sink is much lower than this due to the “hot spot”[50] phenomenon whereby nodes closer to the sink are responsible for handling more data than other nodes further away. In [56] it was shown that the throughput of a 802.11 network, using AODV, is at most equal to $\frac{3}{4}L$. Where the capacity is represented by L and is determined by the following expression $L = (1 - a)L_s$. L_s is the capacity of a link between any two source and destination pair that are more than one hop away from one another. Implying that L represents the capacity between the source node and the node one hop away from the destination (sink) node, where the aL_s represents the capacity of the link between the sink and the node one hop away from it. The authors treat a network, which can be infinitely dense, as one in which a direct path can be

formed between any source node and the sink. A direct path is defined as a linear chain of nodes between the source and destination pair, inclusive of these two, that the source node uses to route its packets to the destination. The number of linear chains between any source and destination pair can be infinite. Furthermore, the distances between each node along a given chain, is assumed not to be constant. Using this reasoning the argument can be made that any network of arbitrarily large size can be treated as the network shown in Figure 3.8.

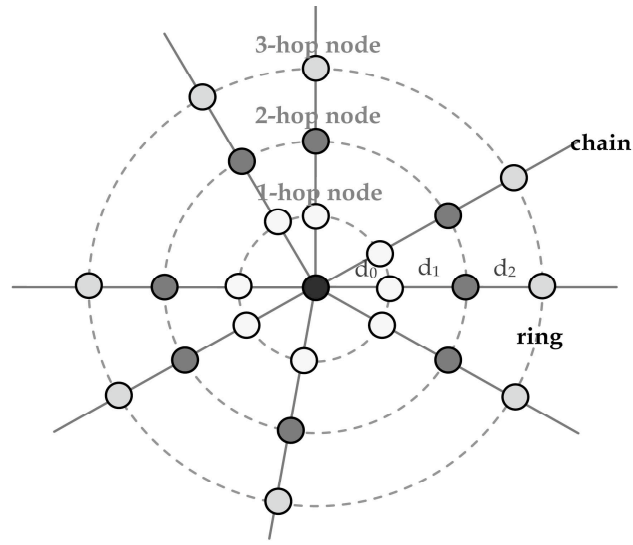


Figure 3. 8 Chain of Nodes

The results presented herein more accurately reflect the throughput of a randomly deployed network. The generalizations in [57], for the capacity of any source and destination pair as a function of the size of the network, are extended to a special case in this work to explain how a network might perform in a real world scenario (e.g. multiple sources and one sink). The results provided are more tangible and can easily be applied in practice. Furthermore the authors show that scheduling is essential to throughput. As the number of nodes per chain increases, the throughput begins to decrease as intuition and the results provided in [57] would suggest. Also as the

number of chains increases it can be seen that the throughput begins to decrease sharply after the number of chains goes beyond 10. Even though the sampling interval for the smart meters will not be as high as those presented in this work, if an increased sampling period $T < 900$ seconds is required to perform real time modeling of the distribution system then smart meter networks will begin to experience bottlenecking.

In order to address this issue several approaches have been suggested to reduce the latency experienced by networks that have certain time constraints. Several authors have implemented a variety of scheduling and rate control algorithms to increase the likelihood with which packets are successfully delivered to their destination. In [61] the authors calculate the maximum sampling period that will allow the network to successfully deliver packets with minimal long run congestion. They develop a packet scheduling algorithm whereby the time optimum number of transmission/reception slots for the nodes in the network is equal to the maximum of twice the number of nodes along the sinks largest branch containing a subtree, and the number of nodes in the network. This assumes that the network to be scheduled is hierarchical, leading to the argument that a single point of failure can jeopardize the survivability of certain parts of the network. In [62] the authors address two different types of congestion that add to overall delays. The first is congestion due to interference generated by neighboring nodes that are transmitting packets at the same time as the transmitting node. The second type of congestion is due to queue (buffer) overflows. They propose an algorithm in which the transmission and generation rate of a node, and of its nearest neighbors are calculated to help control congestion. In the first step each node in the network measures the average rate at which it can process and transmit packets over a predetermined period of time. Second, each node will divide this average rate by the number of nodes relying on it, to route their packets to the sink. Third, each

downstream node will notify the nodes upstream from it about the average rate at which it can process and transmit packets that have been sent it. Using this information along with a fairness design called Probabilistic Sharing and Epoch-Based Probabilistic Sharing the authors were able to show that the number of packets received by each node in the network was approximately equal to the number of packets sent. The network consisted of only 116 nodes, which although is small in comparison, performs well with a constant throughput and no dropped packets when using the Epoch-Based Probabilistic Sharing.

In [63] a platform for real time communication was created using velocity monotonic scheduling (VMS). VMS allows the network engineer to specify a time t with which packets from each source node should reach its destination. It also takes into consideration the physical distance between the source and the destination of the packet as it travels throughout the network. Thereby creating the velocity measurement necessary in order for the packet to reach its destination. Not all packets are processed in the same way however. Those packets requiring a lower latency (i.e. higher velocity) a corresponding monotonic number, much like a label, is attached to the packet. This allows for the prioritization of packets in a network that might have different requirements. As in our application, if the nodes in the network are sending their packets to the same destination then an earliest time-of-arrival first approach is used. Implementation would suggest using priority queues as mentioned in the paper. But a priority queue can be approximated by multiple FIFO queues, with each queue representing a different priority [64]. In addition to this the authors schedule packet transmissions based on the priority queues by manipulating the random back off mechanism in the medium access control layer. Thus allowing nodes that have packets with a higher priority the opportunity to transmit their packets first, while assigning a

later transmission time to those packets with a lower priority. This implies that the throughput of a network will increase and the latency will decrease due to the priority based scheduling. We use a similar approach in our rendition of a quasi-MPLS protocol, however the decision to send the packets with a certain velocity is determined by the sink as opposed to the node itself.

Another popular velocity based routing protocol is [65]. Similar to RAP certain paths are selected based on the need for the packet to reach a certain destination within a certain time. SPEED also uses velocity in order to determine which path a node should send its packet along in order to ensure timely delivery. Each node in the network performs an exponential averaging, or smoothing, of the passed Round Trip Times RTTs for each of its neighboring nodes, closer to the destination than itself, to determine the delay associated with sending a packet to a particular neighbor. The speed at which the neighbor can send the nodes packet is equal to the difference between the nodes distance to the destination and its neighboring nodes distance to the destination divided by the running average of RTTs. The node will select which neighboring node to send the packet to next based on this criterion. Based on a predetermined threshold value for the velocity, the node will select a neighbor that can guarantee that the packet relayed to it will be transmitted at a velocity greater than the threshold. Depending on the application the value can be chosen to select the neighboring node with the highest velocity or it can be selected for load balancing purposes. In which case the velocity will be chosen deterministically for the highest velocity value, and probabilistically for load balancing purposes. When congestion becomes an issue, the algorithm uses a neighborhood feedback loop (NFL) to help determine which set of neighbors it should use to forward its packets. The neighborhood feedback loop is comprised of a local relay ratio mechanism that is used

by each node to determine how many packets it should send and how many packets it should drop, before sending a packet to a neighboring node.

$$r_i = 1 - K \frac{\sum \alpha_j}{\eta_i}$$

For each node i , and all of its neighbors j satisfying the relationship $L_i - L_j > 0$, where, L_i is node i 's distance to the destination and L_j is node j 's distance to the destination, there exists a non empty set of neighbors η_i one hop closer to the destination than node i . For each node i that sends a total number of packets n to a neighboring node j , within a certain time period, the miss ratio is defined as the total number of packets sent to node j that were successfully sent, below a predefined velocity threshold v , and or the packets sent to node j which were lost due to collision divided by n . The node uses this ratio to determine how packets it should send to neighboring node j and how many it should drop. This would occur in a situation where a node is not able to locate a neighbor that can guarantee that the relayed packet will be sent with a velocity greater than or equal to the desired velocity due to congestion. As a result of using this technique, all nodes downstream from any node i are relieved of the congestion. For all nodes upstream from node i the authors implement a back pressure mechanism to relieve congestion. Like RAP, SPEED provides a way for nodes to communicate in real time using velocity as a metric to determine the next best hop towards the destination. This mode of real time communication rests solely on the ability of each node to make a forwarding decision based on the predetermined velocity value. Thus allowing packets to be classified based on their relative importance.

A similar approach has been used with several wireless applications requiring real time requirements [66][67][68][69] using labels along with the packets to make a

decision about subsequent hops. It is similar in the sense that each node makes a decision about each packet that it will send based on the need to have the packet delivered to a particular destination within a certain time. However the difference between this approach and the velocity based algorithms is the processing of each packet. With most of the approaches outlined in [64] and elsewhere in literature, emphasis is placed on how the algorithms behind the decisions that nodes make when forwarding packets as opposed to both the decision making process and the processing of each packet. Packet processing time, depending on the size of each packet or the rate at which intermediate (relay) nodes are receiving packets, can contribute significantly to the overall delay associated with a packet. [70] studies the effect of processing delays on the total end-to-end delay along with: the propagation delay between packets, packet loss rate, number of back offs, and the number of retransmission timeouts between neighboring nodes. In general the total processing time will include the time associated with completing the following tasks. The first component that contributes to the overall processing delay is media access time. In addition to this are the associated queuing delays and the transmission time. The first and last components are a necessary evil otherwise the packet will never reach its intended destination. The solution that they proposed to improve total end-to-end delay is similar in nature to others in the literature whereby the algorithm focuses on the decisions that a node makes, as opposed to how the node operates on the packet itself. The first component is largely dependent on the platform upon which the routing protocol is operating on, and the same can be said for the last component. However, the queuing delays do not necessarily have to depend on the platform used. If the packets received by a node are processed using an approach similar to that of MPLS [71] then only the label must be read in order to make forwarding decisions. Then the packet can be transmitted. This effectively reduces the number of packets

that are queued and left for later, or dropped because of buffer overflow resulting in a potentially longer delay due to packet retransmissions. Furthermore this approach does not require the node to perform any additional calculations in order to determine the next hop. A label is associated with each neighbor that is downstream from each node in the network, and a forwarding decision is based on the priority of the packet as designated by the neighbor upstream from it using the label. The use of labels has been shown to decrease the overall delay and also increase the packet delivery ratio for applications requiring timely delivery of packets [69]. For larger networks, consisting of several thousand nodes, a label-based approach not only provides the aforementioned features listed above, but also has the potential to provide an additional dimensionality to the distribution system [72] if a platform supporting higher data rates were to be used [73] connected in star formation (each meter communicates directly with the access point). As a result the label switching would occur between the sink (access point) and the source (meter). This would be of particular interest to customers in suburban environments looking to solve the last mile problem [73].

3.3 AODV-MPLS Protocol

Fusing both the AODV discovery process and a modified version of MPLS [71], dedicated paths between the sink and a source can be formed and timely delivery of packets can be maintained. As outlined above decreasing the processing time associated with each packet is the focal point of the routing procedure. Like the traditional MPLS protocol, our hybrid protocol uses labels to determine the packets' next hop.

3.3.1 Route Setup and Label Distribution

Route discovery was outlined in the first section of this chapter. The network under consideration is static and each node sends its packets to the same destination, thus eliminating the need for RREQs after the discovery period. RREQs are used primarily to establish a connection between a source and destination pair. Repeated use of RREQs can lead to black hole attacks, which severely decrease the throughput of a network [74]. The route between the source and the destination can be maintained via local Link Status Messages sent between neighboring nodes that have an active route [45] to the destination. As a result the Lifetime field in each node's routing table for the destination can be set to infinity. This creates a dedicated path between the source and the destination. These dedicated paths can be viewed as virtual connections between each source and destination pair in the network. The creation of each path facilitates the use of labels as specified by MPLS. In order for the label switching protocol to be implemented, multiple paths are established corresponding to a particular Forward Equivalence Class (FEC) as specified in [71], and therefore a slight modification of the discovery process is necessary. The modification of the discovery process, discussed herein, also creates multiple paths for intermediate nodes between each source and destination pair as the intermediate nodes setup the forward and reverse paths.

3.3.2 Routing Table Entries

We assume that the nodes closest to the sink are turned on first. Otherwise the nodes

further from the sink would not be able to establish a path to the destination. Subsequent nodes further from the sink are turned on after the nodes' neighbors have begun sending and receiving messages to and from the sink. Nodes in the network establish a route to the sink by broadcasting RREQs just as they are broadcast in the AODV protocol. If an intermediate node has a route to the destination then it will respond with a RREP, otherwise the destination will respond and provide routes not only for the requesting node, but also the nodes forwarding the RREP. In order for this to be accomplished the routing table entries must be modified. Figure 3.2 illustrates the Route Table Entry of the AODV-MPLS protocol. Figures 3.3 and 3.4 illustrate the Route List and the Precursor List for the AODV-MPLS protocol respectively. The Zigbee Route Table entry is shown in Table 3.5.

Table 3. 2 AODV-MPLS Route Table Entry

| <i>Route Table Entry</i> |
|---|
| Destination Address |
| Destination Sequence Number |
| Route State Information and Routing Flags |
| Route List |
| Precursor List |
| Route Lifetime (Expiration time) ∞ |

Table 3. 3 Route List

| <i>Route List</i> |
|--|
| Next Hop ₁ , Hop Count ₁ , Path Cost ₁ , FETTD ₁ |
| Next Hop ₂ , Hop Count ₂ , Path Cost ₂ , FETTD ₂ |
| \vdots |
| Next Hop _k , Hop Count _k , Path Cost _k , FETTD _k |

Table 3. 4 Precursor List

| <i>Precursor List</i> |
|------------------------|
| Precursor ₁ |
| Precursor ₂ |
| ⋮ |
| Precursor _l |

The Zigbee route table entry has similar entries to that of a pure AODV route table entry, with the: destination address, status, group id flag, and next hop address fields being exactly the same. The “No route cache” indicates whether or not a destination node stores source routes; the “Route record required” is used to determine if another route record command frame should be sent to the destination before the next data packet is sent. Source routing is actually necessary in order to send packets in the reverse direction, as will be explained later, but the protocol presents privacy issues and also requires more processing time at each node.

Table 3. 5 Zigbee Route Table Entry

| <i>Route Table Entry</i> |
|--------------------------|
| Destination Address |
| Status |
| No Route Cache |
| Many-To-One |
| Route Record Required |
| GroupID Flag |
| Next Hop Address |

The “Many-to-one” field is used for nodes in the network that serve as concentrators as defined in [46]. The “Status” and “GroupID flag” can be combined into the “Route State Information and Routing flag” field in our modified AODV-MPLS route table entry. The source routing fields in the entry can be removed along with the “Many-to-one” field as well. This would make room for the other necessary fields.

Table 3. 6 Zigbee Route Discovery Table Entry

| <i>Route Discovery Table Entry</i> | <i>Descriptions</i> |
|---|--|
| Route Request ID | A sequence number for a route request command frame that is incremented each time a device initiates a route request. |
| Source Address | The 16-bit network address of the route request's initiator. |
| Sender Address | The 16-bit network address of the device that has sent the most recent lowest cost route request command frame corresponding to this entry's route request identifier and source address. This field is used to determine the path that an eventual route reply command frame should follow. |
| Forward Cost | The accumulated path cost from the source of the route request to the current device. |
| Residual Cost | The accumulated path cost from the current device to the destination device. |
| Expiration Time | A countdown timer indicating the number of milliseconds until route discovery expires. The initial value is <i>nwkRouteDiscoveryTime</i> . |

| | | | | |
|--------------------------|---------------------|-------------------|-----------------------|-----------------|
| Octets: 2 | 2 | 2 | 1 | 1 |
| Frame Control | Destination Address | Source Address | Radius | Sequence Number |
| Network Header | | | | |
| Octets: 0/8 | 0/8 | 0/1 | Variable | Variable |
| Destination IEEE Address | Source IEEE Address | Multicast control | Source route subframe | Frame Payload |
| Network Header | | | | Payload |

Figure 3. 9 General Network Frame Format⁵

⁵ The frame payload field has a variable length and contains information specific to the individual frame types (e.g. Route Request or Route Reply Command Frames)

| | | | | | |
|--------------------------|------------------|---------------------|----------------|--------------|--------------|
| Bits: 0-1 | 2-5 | 6-7 | 8 | 9 | 10 |
| Frame type | Protocol version | Discover route | Multicast flag | Security | Source Route |
| Bits: 11 | | 12 | | 13-15 | |
| Destination IEEE Address | | Source IEEE Address | | Reserved | |

Figure 3. 10 Frame Control Field

| | |
|-------------------------|------------------------|
| Frame Type Value | Frame Type Name |
| 00 | Data |
| 01 | NWK command |
| 10-11 | Reserved |

Figure 3. 11 Values for the Frame Type Sub-Field

| | | | | |
|-------------------------|--------------------------|---------------------|-----------|--------------------------|
| Octets: 1 | 1 | 2 | 1 | 8 |
| Command options | Route request identifier | Destination Address | Path cost | Destination IEEE Address |
| Network Command Payload | | | | |

Figure 3. 12 Zigbee Route Request Command Frame

| | | | | | |
|-------------------------|--------------------------|-----------------------------|-----------|-----------|---------------------|
| Octets: 1 | 4 | 4 | 1 | 1 | 4 |
| Command options | Route request identifier | Destination Sequence Number | Hop Count | Path cost | Destination Address |
| Network Command Payload | | | | | |

Figure 3. 13 AODV-MPLS Zigbee Route Request Command Frame

The values in the Frame Type Sub-Field determine the type of frame that will be used. The NWK command frames include Route Request and Route Reply command frames. We have essentially changed the Zigbee frames to resemble the frame types in

the standard AODV protocol, because it was easier to implement in Opnet. This does not change the results that we have, because the mechanism used to decrease latency is protocol agnostic as the multiprocol label switching name implies.

3.3.3 Timing

Essential to this protocol is timing. Each node that initiates the route discovery process must timestamp the RREQ. Timestamps aide in determining how long it will take a message to reach the destination. Once the RREQ has been timestamped and sent out to discover a route, each node that overhears the RREQ will store the time when it re-broadcasts the RREQ, but will not alter the timestamp field. When the destination receives the RREQ it issues a RREP in the usual manner, and included in the label field of the modified RREP field as shown in Figure 3.8 is where the timestamp is placed. This allows the source node and intermediate nodes that forwarded the RREQ to know exactly how much time it will take to send a message to the destination. In fact, each source node and intermediate node computes a moving average of the four latest timestamp values corresponding with messages that have traveled to the destination regardless of the message type. This value is referred to as the Estimated Time To Destination (ETTD).

This is essential for maintaining up to date timing information between each source and destination pair in the network when parts of the electric distribution system become congested. There are three types of messages that a node can use to update the ETTD field for the destination through a node in the next hop list. The first is a RREP if for some reason the node looses connection with the nodes in the next hop list. The second message type is an acknowledgement (ACK) for a data message that it has sent

to the destination. The third message is a Link Status Message (LSM). A link status message is received from each next hop neighbor after a next hop neighbor receives an ACK or RREP.

In the opposite direction however, the destination stores a running average of the label values for each node in the network to determine exactly how long it takes for a packet to traverse a path to reach the intended source node. Because multiple paths exist between each source node and the destination, and the same paths can be used in the forward and reverse direction, the destination will store a running average of the past four labels for a given node that uses distinct paths between the source and destination.

| Octets: 1 | 1 | 2 | 2 | 1 | 8 | 8 |
|---------------------|--------------------------|--------------------|-------------------|-----------|-------------------------|------------------------|
| Command options | Route request identifier | Originator address | Responder address | Path cost | Originator IEEE address | Responder IEEE address |
| NWK command payload | | | | | | |

Figure 3. 14 Zigbee Route Reply Command Frame

| Bits: 8 | 8 | 64 | 64 | 8 | 20 | 24 |
|---------------------|--------------------------|-------------------------|------------------------|-----------|-----------|-----------|
| Command options | Route request identifier | Originator IEEE address | Responder IEEE address | Path cost | Label | ETTD |
| NWK command payload | | | | | | |

Figure 3. 15 AODV-MPLS Route Reply Command Frame

3.3.4 Forward Equivalence Classes

The premise of this protocol is the FEC, and the mapping of labels to a particular FEC. A FEC consists of a group of packets, each of which is forwarded along the same route. Unlike in the MPLS standard where FECs are largely based on the network

addresses of routers in a network, the FEC in our protocol is based on the timing information described above. MPLS provides a form of Quality of Service, whereby each node in the network will guarantee that only a certain amount of time will be spent processing and forwarding the packet towards its destination. Applications that currently utilize this concept are: Voice over IP (VoIP), online gaming, as well as IP-TV. Although the applications considered herein do not require the same type of service, larger networks with stringent time requirements would benefit from a similar approach.

A node that generates a packet is referred to as an ingress node; the node for which this packet is destined is called the egress node. The nodes between these two nodes are called label-shifting nodes. They perform a label shifting operation to ensure that the packet is routed along the correct path to the destination in the forward direction, and to the correct source node in the reverse direction. A connection established between a source node (ingress node) and the destination node (egress node) is referred to as a label shifted path. Thus each ingress node, selects a path with an estimated latency that is less than or equal to the amount of latency it is willing to tolerate. The ingress node then places all packets with similar latency requirements into the same FEC. This is equivalent to saying that all packets that traverse the same path with a latency, D_{x_i} , equal to some estimated amount, γ_l , will belong to the same set X . The set X_i represents the i^{th} Forward Equivalence Class that the packets x_{ik} belong to.

$$X_i = \{x_{i1}, x_{i2}, \dots, x_{ik}\} \text{ for } D_{x_i} = \gamma_{x_i} \quad \forall i = 1, \dots, m; k = 1, \dots, p$$

The index k represents the number of packets that belong to any given FEC, thus the number of packets from FEC to FEC need not be the same.

3.3.5 Labels

A label is a fixed length packet identifier used between two adjacent nodes. The label assigned to a particular packet represents the Forward Equivalence Class that the packet belongs to. The label not only uniquely identifies the FEC that a packet belongs to, but it also uniquely identifies which neighboring node the packet came from. Labels drive the routing decisions that are made at each node. Because the label uniquely identifies the neighbor from whence a packet comes from, as well as the FEC that it belongs to, the recipient of a packet need only look at the label to determine which neighbor sent the packet. The recipient then determines the next hop for the packet based on the FEC that the packet belongs to, which is determined by the label attached to the packet. The node removes the label from the packet, and then attaches another label to the packet corresponding to the next hop in the FEC. Then the node forwards the packet to the next hop and the process continues until it reaches its destination. This tremendously reduces the amount of time spent processing each data packet. Figure 3.9 illustrates the general network frame format. The frame can have as little as $64 + n$ bits where n represents the variable size of the payload. This is only possible if the: IEEE source and destination addresses, the source route subframe, and multicast control subframe fields are removed. Conversely the frame can include as many as $136 + 16(m + 1) + n$ bits with the first term accounting for all terms excluding the variable payload and source route subframe fields. Both of which are accounted for by the variables n and m respectively. The additional 16 bits in the second term represent the control information included in the source route subframe field as illustrated in figure 3.16.

| | | |
|------------------|-------------|-----------------|
| Octets: 1 | 1 | Variable |
| Relay Count | Relay Index | Relay List |

Figure 3. 16 Source Route Subframe

As will be explained in Section F, source routing is one of two options that have to be used in order to route packets from destination to the sink if a label shifting technique is not used. As a result this drastically increases the amount of unnecessary information needed in order to route a message from the destination to a source node. With a label solution each node only reads the 24-bit label that is attached to the leftmost field of the network header to determine the next hop in the FEC. The label corresponding to the node in the next hop list of the FEC is attached to the packet and then retransmitted. This reduces the overhead associated with packet processing, thereby decreasing the end-to-end latency.

3.3.6 Forward Path (Toward Sink)

There are two scenarios that must be addressed for forward path setup. The first scenario is one in which a source node requests a route to the destination, and none of the intermediate nodes that overhear the RREQ are able to accommodate the request (i.e. the intermediate nodes do not have a path to the destination). In such a situation only the destination may respond.

In the second scenario there could be an intermediate node, or a collection of intermediate nodes, that have a route to the destination. In which case the same procedure is used, except that the intermediate node(s) generate a RREP and the label in the timestamp field of the RREP is based on the responding nodes most recent ETDD value.

First it is worth mentioning that if the correct transmission radius is selected for the nodes in the network, each node will have at least k neighbors. This radius also guarantees K-connectivity as the number of nodes in the network increases [75]. The unique property of K-connectivity is that, there exist at least k node disjoint paths between all pairs of nodes. This property follows from Menger's Theorem, which states that when no $k-1$ vertices can be removed to disconnect the graph, there exist at least k vertex-disjoint paths between all pairs of nodes. Although this fundamental result holds true for a sizeable number of nodes in a network, the paths must be established and this depends on the underlying routing protocol being used.

The radius for each node in the network is approximately $r = .244954$ times the length of one side of the square. Which in our case is 1500 meters. This value is an estimate because the expression used to calculate this value is not an exact formula. It is an expression that satisfies certain properties needed in order to prove K-connectivity for a randomly distributed set of vertices on a graph. The value of the radius is based on a number of factors as outlined in [75], one of which is obviously the number of neighboring nodes that any one node is connected to. It has been shown that more than $k=3$ paths provide little to no additional throughput [76][77] for reactive routing protocols. However as mentioned before the routing protocol is responsible for the development of these paths, and the AODV-MPLS protocol generates more routes than k link disjoint paths as guaranteed by K-connectivity results. These additional routes are not node disjoint, and could possibly share similar links with other routes. Unlike in [76][77][45] our objective, in addition to providing timely delivery of packets, is to also ensure that there are enough paths between any source and destination pair in case of link or node failures.

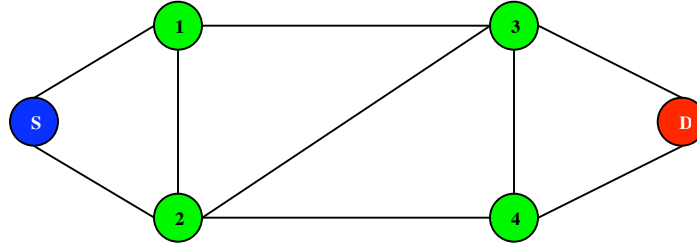


Figure 3. 17 Example Network

In the first scenario a source node requesting a route to the destination will broadcast a RREQ addressed to the destination. Each intermediate node overhears the RREQ will rebroadcast it until the RREQ reaches its intended destination. Once the RREQ reaches its destination the destination sends a RREP to the nodes that it overheard the RREQs from. In a case in which the destination has more than $k=3$ neighbors it will only respond to 3 RREQs arriving from 3 distinct neighbors. Unlike the regular AODV protocol and the protocol described in [76], in our protocol RREPs are only forwarded to neighboring nodes that it overheard RREQs from that have a hop count, *towards the source*, that is less than or equal to their own. The hop count is ascertained from the RREQ messages. When a node receives a RREQ from neighboring upstream nodes, it will insert into the precursor list each of the neighboring upstream nodes and will not the hp count in the RREQ message corresponding to the neighboring node it overheard the RREQ from. As the RREQ propagates along the different edges to different vertices, the nodes located at these vertices will overhear all neighboring RREQs. Consider node 2 in Figure 3.13. It will overhear RREQs from nodes: S, 1, 3, and 4. The hop count in the RREQs overheard from node S and 1 are 1 and 2 respectively. This means that nodes S and 1 are 0 and 1 hop away from the node S respectively; and that node 2 is 1 hop away from node S. Therefore when a RREP arrives at node 2, from a downstream node, node 2 will send the RREP to both nodes because the hop counts recorded in the RREQs are less than or equal to its own hop

count towards the source. Node 1 will do the same as well. Nodes 1 and 2 will not respond to the rebroadcasts of nodes 3 and 4 because the hop counts in both of these messages will be 3, implying that both of these nodes are 2 hops away from the source. The hop count recorded in these RREQ messages suggests that the nodes are closer to the destination than node 2 and therefore it would be counterproductive to send a RREP to these nodes as it will result in a loop.

RREPs are sent to the nodes in the same order in which a node overhears RREQs from neighboring upstream nodes. As an example if node 3 receives consecutive RREQs from nodes: 1, 2, and 4 respectively, then it will respond to nodes: 1, 2, and 4 with 3 consecutive RREPs accordingly. Assuming that node 2 overheard two RREQs, one from node S first and another from node 1 it will respond to these two nodes with two RREPs. One corresponding to the RREP generated by node 3 and another one corresponding to the RREP generated by node 4. The first RREPs received from unique neighboring downstream nodes are forwarded to each of the recipient node's neighboring upstream nodes. Returning to the example, nodes S and 1 will have two node disjoint routes to the destination (node D). The first RREPs received from neighboring downstream nodes, are forwarded back towards the source node just as in the AODV protocol. The only difference now is that each node will receive multiple routes to the destination through each of its neighboring downstream nodes. It might not be clear why node 4 receives a RREP from node 3, but the reason for doing this is to give node 4 an alternative path to the destination if the link between itself and the destination should fail. Furthermore if node 4 should receive a packet from a node upstream from it, and the link between nodes 4 and D fail, instead of issuing an error message to the node that sent the packet, node 4 can simply forward the packet to the destination via node 3. The same would hold true, in the opposite direction, for

packets traveling to the destination via node 3 if the link between the destination and node 3 is severed. As nodes receive additional RRES they are forwarded to their neighboring upstream nodes. However unlike before the neighboring upstream node will not forward the additional RREPs that it receives but rather use the routes provided by the RREPs for itself. As an example, because nodes 3 and 4 are the same distance away from the source as indicated by the first RREQs that they receive from nodes 1 and 2 respectively, they will send each other RREPs. However a mechanism must be put in place to ensure that a loop does not develop between these two nodes. In a situation in which the distances to the source is the same for neighboring nodes, whenever one of the two neighboring nodes receives a packet from a neighboring upstream node and forwards it to its neighboring node, with the same distance to the source node, the receiving node is only permitted to forward it to a neighboring node with a greater distance from the source. So instead of node 3 resending a packet that it has received from node 4 back to node 4 it will send it to the destination. As an example consider node 4 when it receives a packet from node 2 and decides to send it to node 3 which is the same distance away from the source as is node 3. When node 3 receives the packet it will send it to a node with a greater distance away from the source, which in this case would be node D. Node 3 determines this because node D will respond to node 3s RREQ with a RREP which would indicate that node D is node 3s next hop downstream neighbor, and therefore would be a greater distance away from the source than any other node along the path that the RREQ has traveled. As a result node 3 would forward a packet that it receives from node 4 to node D. The same holds true for packets being forwarded to node 4 from node 3.

After the route discovery process is completed, in our example above, nodes 3 and 4 will have 2 paths(routes) to the destination as explained before. Nodes 1, 2, and S have

4, 5, and 4 routes to the destination respectively, in accordance with the rule for forwarding and storing routes as explained above. These different routes, depending on the latency associated with each one, form the set of FECs described in section C. Once the routes have been obtained, and the FEC has been selected, the nodes will attach the label shown in Figure 3.17 to the network header shown in Figure 3.8 and the data message will be sent. Instructions for reverse route setup as well as instructions explaining how nodes obtain timing information for the different FECs are explained in the next section.

3.3.7 Reverse Path (Toward Source)

Whenever a node sends or receives a message in the AODV-MPLS protocol, a local timestamp is created. Depending on the message type, the timestamp may or may not be included. We require that timestamps be used, so that packets destined for a source node originating at the destination can be routed with minimal processing time. The MAC common part sublayer (MCPS) of an IEEE 802.15.4 device creates a timestamp with 24 bits of precision, each time a packet is sent or received [46].

Consider, once again, the example network in Figure 3.16. Let us examine the process of label distribution, during the discovery phase. With node S being the source of the route discovery process, the RREQ will propagate through the network as usual and the different paths are set up according to the procedure outlined in section E. Each node that receives the RREQ will create a local timestamp corresponding to the time when the RREQ was rebroadcast. Once the RREQ reaches the destination via nodes 3 and 4, a RREP is generated and the timestamp corresponding with the reception of the RREQ is included in the label field of the RREP and sent back to node S via nodes 3

and 4. As the RREPs propagate through the network to node S, as outlined in section E, each node creates another local timestamp corresponding with the reception of the RREP. The forwarding also timestamps the RREP before unicasting it to the nodes in the precursor list. In addition to these two timestamps, another local timestamp is created that corresponds with the time in which it overhears the nodes in the precursor list unicast the RREP to the nodes in their precursor list. This is done so that each node can verify that the labels being sent to it are from the neighboring upstream nodes that it received RREQs from and sent RREPs to. As an example consider nodes D and 3. When node D generates a RREP and unicasts it to node 3, node 3 will know exactly how long it took for the RREQ to reach the destination because label field in the RREP contains the timestamp corresponding with the RREQs arrival. We assume that the processing time between reception of the RREQ and the transmission of the RREP is negligible. When node 3 unicasts the RREP to the first neighboring upstream node that it received a RREQ from, which we have not designated for this example, it will timestamp the RREP and then transmit it to this node. Node D will overhear the transmission and will create a local timestamp corresponding with the arrival of the packet. Therefore node D knows exactly when the packet arrived at node 3, if we assume negligible processing time. This value is also the value that will be used as a label when sending packets to node D. Node D records this value in the precursor list entry for node D. The process continues as such until all nodes in the network forwarding RREPs, know exactly when their upstream neighboring nodes have received the forwarded RREP. It should be noted that nodes without any precursors, or upstream neighbor nodes, will have to send a link status message to each of its next hop neighbors when it receives a RREP to indicate when the next hop neighbors RREP was received. In the example just discussed that would be node S.

Next we demonstrate how a packet travels through the network. We consider the path: S, 1, 2, 3, D. First we introduce some notation.

$t_{i \leftarrow j} :=$ The time at which a packet/message arrives or is completely received at node i from node j

$\delta() :=$ The shift operator

$EXP_l :=$ The experimental field attached to the packet at node l

$i \rightarrow k :=$ Packet sent from node i to node k

$L_{i \rightarrow k} :=$ Label attached to a packet sent from node i to k

For

$$\begin{aligned} S \rightarrow 1 : L_{S \rightarrow 1} &= t_{S \leftarrow D(1,2,3,D)} \\ 1 \rightarrow 2 : L_{1 \rightarrow 2} &= t_{1 \leftarrow D(2,3,D)} = \delta(t_{S \leftarrow D(1,2,3,D)}); EXP_1 \\ 2 \rightarrow 3 : L_{2 \rightarrow 3} &= t_{2 \leftarrow D(3,D)} = \delta(\delta(t_{S \leftarrow D(1,2,3,D)})); EXP_2, EXP_1 \\ 3 \rightarrow D : L_{3 \rightarrow D} &= t_{3 \leftarrow D} = \delta(\delta(\delta(t_{S \leftarrow D(1,2,3,D)}))); EXP_3, EXP_2, EXP_1 \end{aligned}$$

Node D now has a 24-bit label along with a stack of experimental field values. The label is actually the timestamp value that node D overheard from node 3 when node 3 unicasted the RREP to the first precursor node that it overheard the RREQ from. The destination will perform an inverse shift operation using: EXP_3, EXP_2, EXP_1 to determine: $t_{2 \leftarrow D(3,D)}$, $t_{1 \leftarrow D(2,3,D)}$, and $t_{S \leftarrow D(1,2,3,D)}$ respectively.

For each packet that journeys through the network the same procedure is used. When the destination needs to send a message to a node in the network it simply takes the complement, either 1s or 2s complement, of the label associated with the source node that it wants to send a message to, then attaches the experimental bits to the label and then sends it to the source node through the neighbor whose label corresponds with the source node. As an example, consider what would happen if node D must send a

message to node 1 along the path: D, 3, 2, 1. Node D will send $L_{3 \rightarrow D}^C$; EXP_3, EXP_2, EXP_1 to node 3. Node 3 will perform the same inverse shift operation that node D did when it received a packet from node 3, then it removes EXP_3 from the stack and then sends the message to node 2. The process continues until node 1 receives the message.

The complement is use so that forwarding nodes along the reverse path know exactly which precursor node to send the message/packets to, to ensure that the packet is successfully routed to the intended destination.

Each inbound label maps directly to an outbound label at a given node, along the forward path. The same holds true along the reverse path. Once each node has determined what their inbound and outbound labels are, the complement of the labels is taken and stored in the precursor and next hop lists entries that correspond with the respective neighboring nodes. For example, node 2 will have stored two inbound labels for node 1 and two inbound labels for node S according to the rules outlined in section E. It will also have two outbound labels for node 3 and two for node four. Node 2 will also store the complements of these labels for nodes S and 1 in the precursor list and nodes 3 and 4 in the next hop list. When node 2 receives a packet from node 3 or 4, with one of the labels stored in the next hop list attached to the packet along with the EXP stack, it will know exactly which label in the precursor list it should select. It will then remove the top most EXP field value, and then attach the label to the packet along with the remaining EXP stack and forward the packet. The complement of the forward path inbound labels become the outbound labels along the reverse path. The complement of the outbound labels along the forward path, become the inbound labels along the reverse path. It should be noted that the size of the

experimental field should be eight bits in length and be an integer value with a range of 0x00-0xff. This corresponds with 256 different octet values each of which corresponds with a different second value. Therefore approximately 4.26667 minutes can be expressed by the values in the range specified above. Depending on the application the range may vary. If the amount that a label must be shifted by requires less than eight bits, then the node performing the shift will only use the number of bits necessary to express the shift in value. As an example, if node 2 receives a packet from node D at some time t_2 such that node 2 only requires 3 experimental bits to perform the operation $L_{2 \rightarrow 3} = t_{2 \leftarrow D(3,D)} = \delta(L_{1 \rightarrow 2} = t_{1 \leftarrow D(2,3,D)})$, then only 3 experimental bits need be sent. Therefore the stack will be variable in size depending on the amount of time that it takes for packets to travel between neighboring nodes.

An explanation of the shift operation is warranted. By shift we actually mean that the next number in a 24-bit series of numbers is selected. For example if node 2 receives a packet at some time t_2 from the destination and node 1 receives a packet from the destination through node 2 at some time $t_1 > t_2$, when node 2 receives node 1's label $L_{1 \rightarrow 2} = t_1$, node 2 will shift the value of this label as explained above to resemble the time t_2 . Say for instance node 2 forwards a packet to node 1 from the destination, and it is received at times:

$$t_2 = 0000000000000000100000000; t_1 = 0000000000000000100001010.$$

Table 3. 7 Global Network Time

| Time |
|------------------------------|
| \vdots |
| 00000000000000000100000000 |
| 000000000000000000100000001 |
| 0000000000000000000100000010 |
| 0000000000000000000100000011 |
| 0000000000000000000100000100 |
| 0000000000000000000100000101 |
| 0000000000000000000100000110 |
| 0000000000000000000100000111 |
| 0000000000000000000100001000 |
| 0000000000000000000100001001 |
| 0000000000000000000100001010 |
| \vdots |



Figure 3. 18 AODV-MPLS Label and Experimental Field⁶

The difference in time is ten seconds. Table 3.7 demonstrated how node 2 would have to shift the timestamp/label that it receives from node 1 by ten binary values back in time. In this example the experimental field would require four bits to express this difference. This process makes the forwarding process much easier along the forward and reverse paths.

When a node must forward acknowledgements(ACK) from the destination to nodes in the network, without the use of labels, it must record the node in the precursor list that it received the packet from along the forward path, as well as the address of the node. This would require additional memory for nodes closer to the destination, especially as the density of the network increases, thereby changing the requirements of meters closer to the destination. It also requires forwarding nodes to spend more time going

⁶ EXP represents the entire EXP stack.

through lookup tables to determine the precursor node corresponding to the source node for which the packet is destined. The use of labels decreases this unnecessary processing time.

If source routing is used the network layer header contents must be examined first to ensure that a route table entry exists for the source address [46] in the source routed data frame. Source routing allows any node in the network to locate other nodes in the source route list. This information can be used as a vector for the dissemination of data mining code/software capable of extracting personally identifiable information about any node included in the source route list.

In either case, if source routing is used or if each intermediate node records the addresses of the source nodes for which it is forwarding packets, intermediate nodes have access to node addresses and the location of these nodes. If the label stack solution is used instead, the privacy of the user is preserved so long as each node is only capable of reading the topmost label. A Trusted Platform Module approach to ensuring that each node in the network only has access to the information needed relevant to route a packet can guarantee the privacy of any given node in the network [Trusted Computing Group Architecture Overview]. This is similar in essence to Onion Routing whereby each node adds and removes additional bits (i.e. layers) to and from a packet respectively as it is routed through the network to the destination. It has been shown that protocols with this property guarantee anonymous routing among the nodes in the network, and protects against potential eavesdroppers [78]. Therefore the use of a label stack consequently provides a way for any two nodes in a network to communicate anonymously.

3.4 Results and Analysis

As intuition would suggest the average delay for the AODV-MPLS protocol is on average 300 milliseconds faster than the regular AODV-Zigbee protocol, whilst the difference in throughput of the two protocols, was on average 1.5 packets.

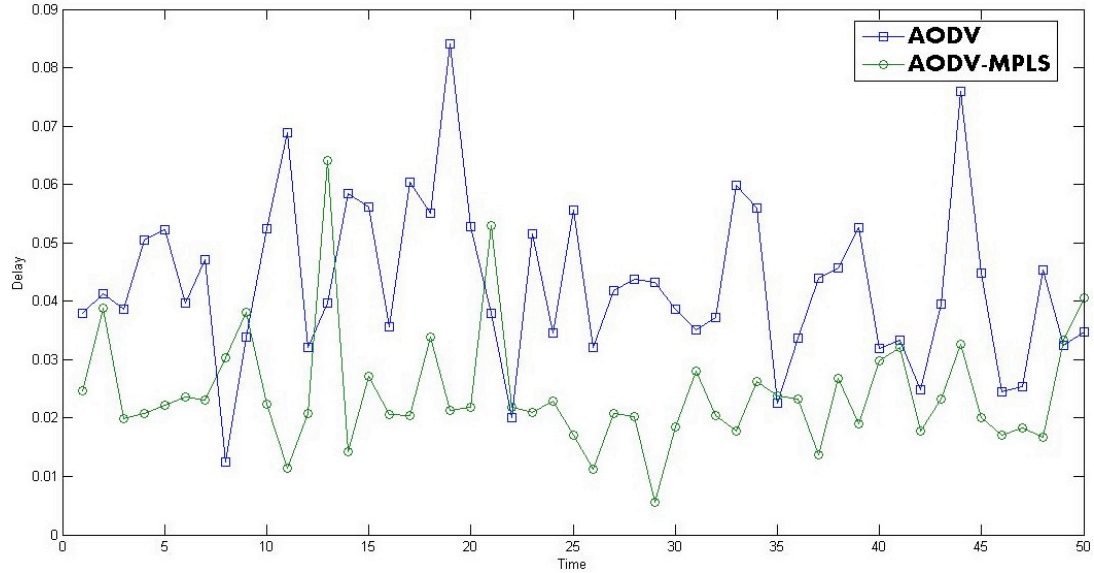


Figure 3. 19 Delay (seconds) vs Time (seconds) – 48 hour period

To simulate the two different protocols we used the Opnet discrete even simulation engine. We simulated a network of 100 nodes on a 1500meter by 1500meter grid with a node placement distribution following a uniform random variable in both the vertical and horizontal dimensions. These dimensions were chosen based on Census data. The average number of homes per square mile in the United States is roughly 33, which is roughly 33 homes per 2600 kilometers squared. We reduced the dimensions and increased the number of nodes on the grid to resemble a more densely populated neighborhood where such a technology would be used. The size of the square grid was based on census data for the state of New York where on average there are roughly 162 homes per 2600 kilometers squared. We used 130 homes per 2600 kilometers squared, which is exactly 100 nodes per 2500 kilometers squared. This number was

chosen at random and has no particular significance.

Ninety-nine of the one hundred nodes send packets to the destination node, node 81, located in the middle of the network as shown in Figure 3.20 every 15 minutes. We ran 50 different simulations with the aforementioned sampling period to obtain the results in figures 3.91 and 3.21. Every nine seconds a different node sent a message to the destination. The size of the packet was determined by a pilot program in Austin Texas, [59] in which 500,000 meters reported 15-minute interval data over the course of a year. Four hundred MB of data were generated which corresponds with approximately 11 KB of data every 15 minutes. The transmission radius for each node was set at a maximum of, 250 meters, and the data rate was set to 250kbps as outlined in the IEEE 802.15.4 standard. The maximum transmission radius was selected to create $k=3$ node disjoint paths. The other Zigbee and IEEE 802.15.4 specifications outlined in [46] and [72] are included in the Opnet discrete event simulator.

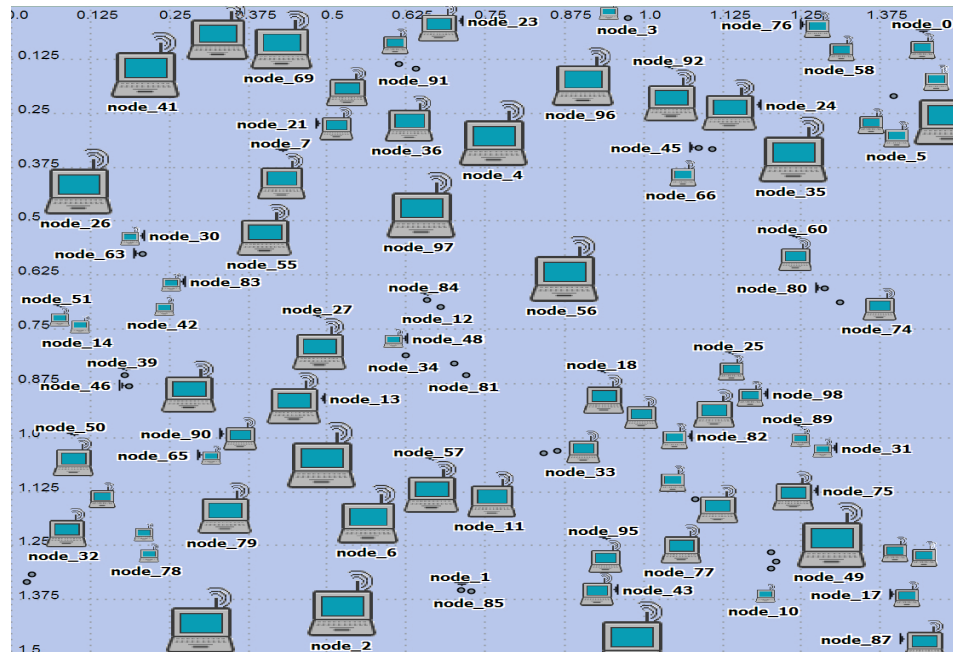


Figure 3. 20 100 Node Network

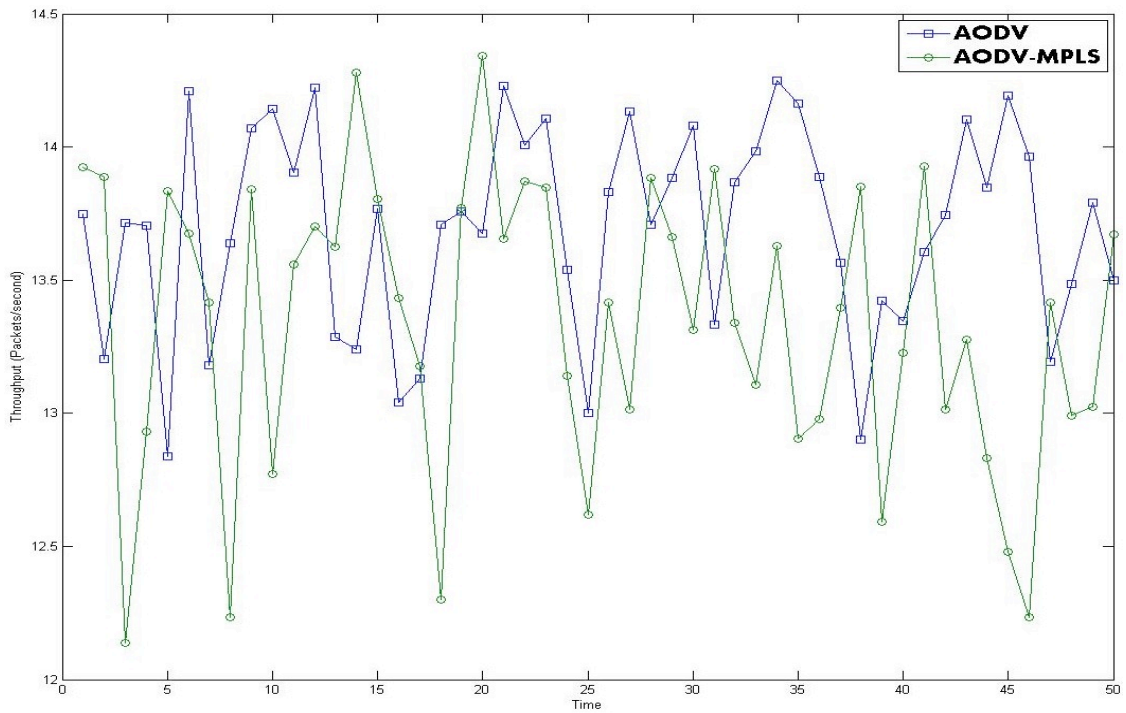


Figure 3. 21 Throughput (packets/second) vs Time (second)—48 hour period

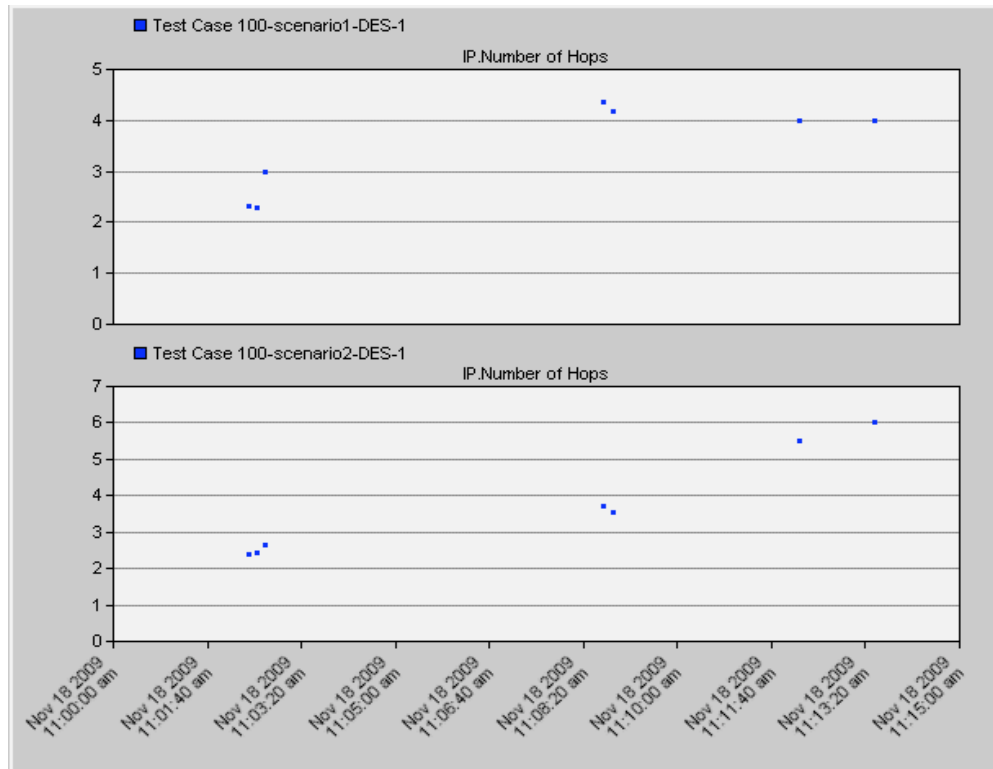


Figure 3. 22 Number of IP Hops for AODV(top), AODV-MPLS(bottom)

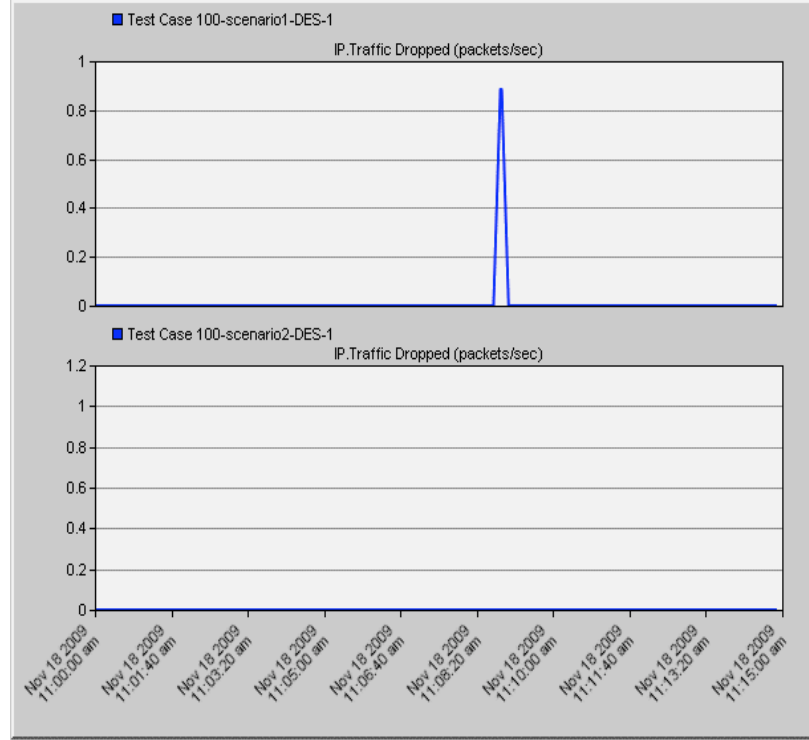


Figure 3. 23 Number of Dropped Packets AODV(top), AODV-MPLS(bottom)

The results in Figure 3.23 show that on average the number of dropped packets is approximately the same. However as time progresses, over the 15-minute period, it was noticed that towards the 10 minute mark the AODV protocol repeatedly suffered a surge in dropped packets. This is due to the fact that dedicated paths are not created between the source nodes and the sink node, with the AODV protocol. As a result repeated route discovery will take place whenever a node believes that one of its neighbors has moved outside of its communication radius. Because the network is static the need for repeated route discovery is not necessary. However with a pure AODV routing protocol if a node does not receive either a control or data packet within a certain interval of time, ≈ 3000 milliseconds, it will reinitiate the route discovery process.

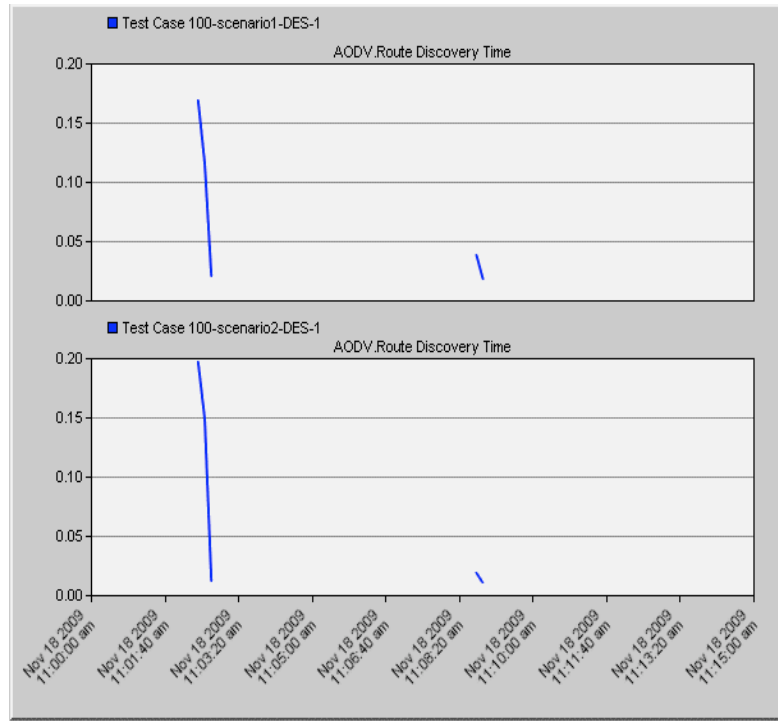


Figure 3. 24 Route Discovery Time

Figure 3.25 show most nodes in the network were able to discover a route to the destination approximately 3 simulations minutes after the beginning of the simulation. The AODV-MPLS protocol took slightly longer than the AODV protocol to complete the route discovery phase. This implies that the protocol was able to provide more routes to the destination for each node than AODV during the same time period. As can be seen between 11:08 and 11:10 the AODV protocol reinitiates the route discovery period to accommodate the nodes that it was not able to create routes to the destination for earlier. Although there is an increase in the setup time for the AODV-MPLS protocol it is only a difference of 50 milliseconds.

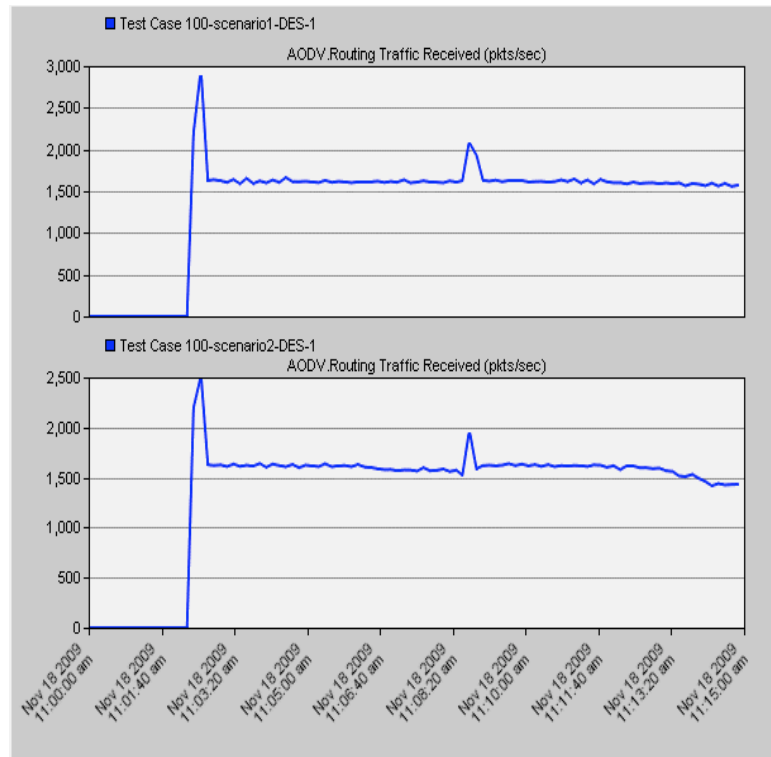


Figure 3. 25 Received Routing Traffic

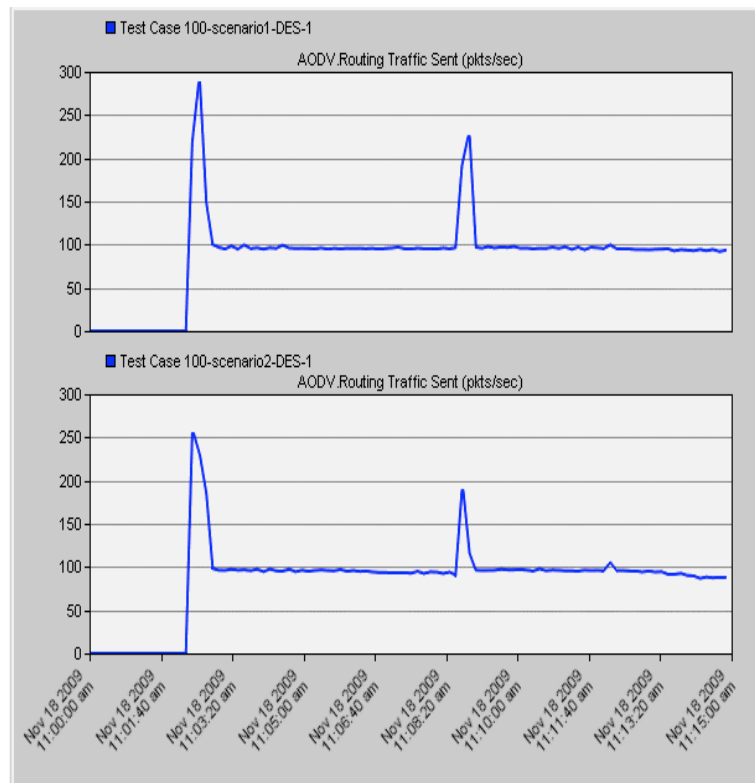


Figure 3. 26 Routing Traffic Sent

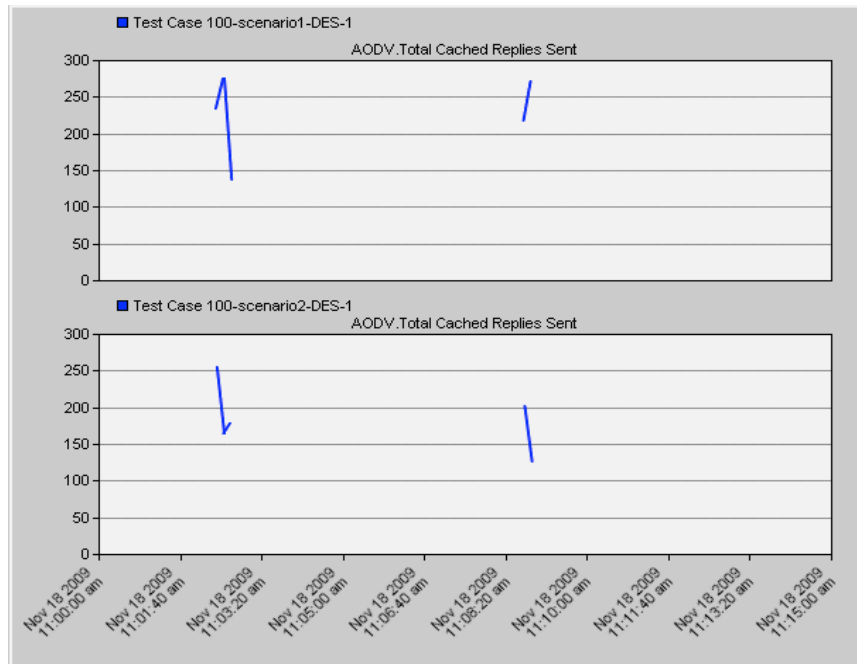


Figure 3. 27 Total Cached Replies Sent

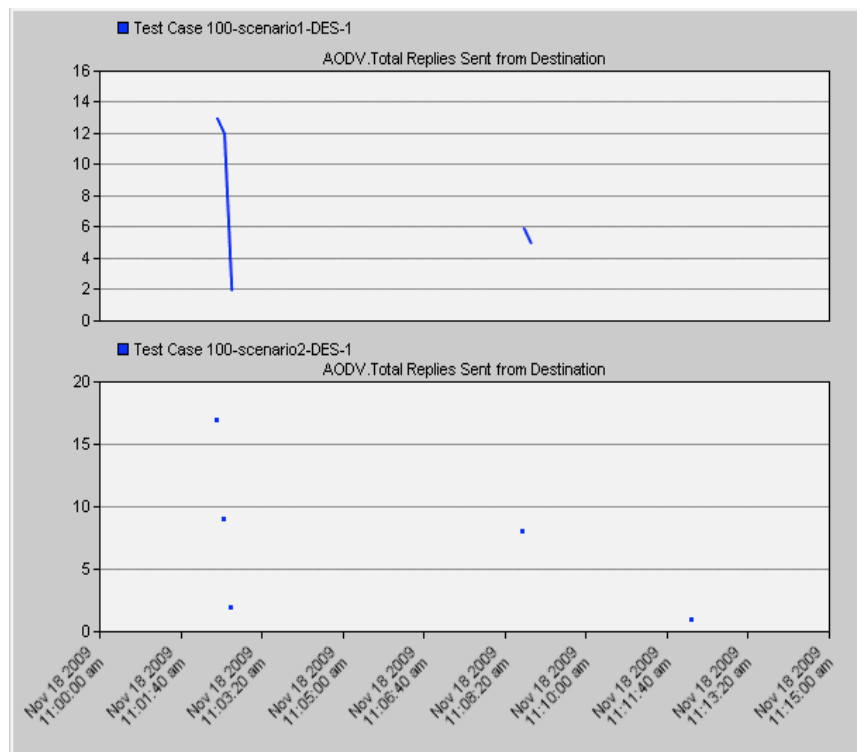


Figure 3. 28 Number of Replies Sent from Destination

Figures 3.26 and 3.27 demonstrate how AODV-MPLS generates the same amount of routing traffic as AODV, and this can be attributed to the fact that the AODV-MPLS protocol does not rely solely on the destination to provide routes to the nodes in the network, as can be seen in Figure 3.29. Given the route setup process of the AODV-MPLS protocol one might think that the number of control packets used to create the paths would be greater. This is not the case though, because the AODV-MPLS protocol relies on the paths created by intermediate nodes—Figure 3.28. Comparing figures 3.28 and 3.29 it is apparent that the AODV protocol uses cached routes at intermediate nodes just as much as it uses routes created by the destination. However AODV-MPLS relies more heavily on the cached routes to create routes between the source nodes and the destination. The destination node in the AODV-MPLS scenario generates RREPs at 5 different points in time in comparison to the AODV protocol, which generates a stream of RREPs for a period of roughly 45 seconds. So even though AODV-MPLS protocol requires more control information (i.e. labels inserted into each RREP) because the intermediate nodes respond the *number* of control packets being sent (Figure 3.27) and received (Figure 3.26) is roughly the same.

CHAPTER 4

LOOKING FORWARD & CONCLUSION

4.1 Conclusion

We have successfully shown that a multipath, privacy preserving, routing protocol can be developed by exploiting the route discovery phase to create dedicated paths between each source and destination pair. As a result some of the more complex schemes and algorithms developed in [61][63][65] are not necessary. Changing the way packets are read alone reduces the amount of time that a packet spends traversing the network by almost half a second. As the network scales, timing will become even more critical. This solution only provides fast and reliable delivery for networks of a reasonable, size. Beyond a certain point the results obtained here will be much like the results for the regular AODV protocol. That being said another approach to solving the problem of data extraction for smart meter data must be considered. We have made one such recommendation.

If a wireless solution is going to be used, a different network configuration is needed altogether. If instead of using a mesh network, a star network were used, this would eliminate the need to use complicated and memory intensive routing protocols to ensure successful delivery of data. Furthermore with a star configuration decreases the surface area of attack. With a mesh network depending on how crafty a malicious agent is, he or she would have the ability to learn personally identifiable information about their neighbors. Removing the ability to have one customer's house speak with another customer's provides a level of security and privacy that currently does not

exist. There is only so much security that can be used to secure a network with nodes that function as relays for other nodes in the network. Thus warranting this alternative solution.

4.2 Looking Forward

The information that is collected by these meters provides a wealth of information, which if used properly, provides benefits to both the customer and the supplier. However if mishandled by the utility customer anonymity and privacy will be no longer. To this end we would like to determine just how much information is exactly necessary in order for a utility to provide real time pricing options to their customers and also maintain the health of the distribution infrastructure. Currently most deployments in the United States that have smart meter programs in place poll the meters every 15 minutes. However some say that each home should report a value at much higher frequency in order for there to be any real benefit to the customer. Because of the ambiguity surrounding this issue, as well as the inherent security and privacy concerns associated with the current set of solutions, a much more thorough investigation is warranted in order to fully understand how to provide both parties with a meaningful experience that will be beneficial to both sides.

BIBLIOGRAPHY

[1]<http://www.cnn.com/2010/TECH/05/03/smart.dust.sensors/index.html?iref=allsearch>

[2] Mus˘aloiu-E., R˘azvan, et al “Life Under your Feet: A Wireless Soil Ecology Sensor Network”

[3] Allen Y. Yang , Sameer Iyengar, Shankar Sastry, Ruzena Bajcsy, Philip Kuryloski, Roozbeh Jafari. Distributed segmentation and classification of human actions using a wearable sensor network. In *Proceedings of the CVPR Workshop on Human Communicative Behavior Analysis*, 2008

[4] Khan, A., Jenkins, L.: Undersea wireless sensor network for ocean pollution prevention. In: *Proc. 3rd Int’l. Conference on Communication Systems Software and Middleware (COMSWARE 2008)*, pp. 2–8. IEEE, Los Alamitos (2008)

[5] Bennett C., Wicker B. S., Cardell J., “Residential Demand Response Wireless Sensor Network.” *Fourth Annual Carnegie Mellon Conference on the Electricity Industry: Future Energy Systems-Efficiency, Security, and Control 2008.*, Pittsburgh, Pennsylvania, 2008

[6] Sam Phu Manh Tran , T. Andrew Yang, Evaluations of target tracking in wireless sensor networks, *Proceedings of the 37th SIGCSE technical symposium on Computer science education, March 03-05, 2006*, Houston, Texas, USA

[7] K. Bult , A. Burstein , D. Chang , M. Dong , M. Fielding , E. Kruglick , J. Ho , F. Lin, T. Lin , W. Kaiser , H. Marcy , R. Mukai , P. Nelson , F. Newburg , K. Pister , G. Pottie , H. Sanchez , O. Stafsudd , K. Tan , S. Xue , J. Yao, Low power systems for wireless microsensors, *Proceedings of the 1996 international symposium on Low power electronics and design*, p.17-21, August 12-14, 1996, Monterey, California, United States

[8] Jason Hill , Robert Szewczyk , Alec Woo , Seth Hollar , David Culler , Kristofer Pister, System architecture directions for networked sensors, *Proceedings of the ninth international conference on Architectural support for programming languages and operating systems*, p.93-104, November 2000, Cambridge, Massachusetts, United States

[9] S. Roundy and P.K. Wright, A piezoelectric vibration based generator for wireless electronics, *Proceedings of Smart Mater. Struct.*, vol. 13 (2004), pp. 1131–1142.

[10]<http://www.newscientist.com/article/dn11907-superconducting-power-line-to-shore-up-new-york-grid.html>

[11] <http://www.electricdistribution.ctc.com/microgrids.htm>

[12] Stanislaw A. Joseph, Climate Changes Everything: The Dawn of the Green Economy, Deloitte Energy Center Report

[13] <http://www.google.com/corporate/green/footprint.html>

- [14] California Senate Bill No. 17, Chapter 327
- [15] UtilityAMI 2008 Home Area Network System Requirements Specification
- [16] <http://www.homeplug.org>
- [17] Katar S., Krishnam M., Newman R., and Latchman H., Harnessing the potential of powerline communications using the HomePlug AV standard., In *Broadband Technology*, August 2006
- [18] <http://www.smarthomeusa.com/info/x10theory/>
- [19] <http://www.insteon.net/>
- [20] <http://www.echelon.com/communities/energycontrol/developers//lonworks/>
- [21] <http://www2.clipsal.com/cis/technical/>
- [22] http://www.homegridforum.org/news_events/pr/02_25_09/
- [23] <http://www.upapl.org/>
- [24] Y. Kato, T. Ito, H. Kamiya, M. Ogura, H. Mineno, N. Ishikawa, T. Mizuno, "Home Appliance Control Using Heterogeneous Sensor Networks," *IEEE Consumer Communications and Networking Conference (CCNC2009)*.

- [25] Jian-she Jin., Jing Jin, Development of Remote-Controlled Home Automation System with Wireless Sensor Network, *Fifth IEEE International Symposium on Embedded Computing*.
- [26] T. Haenselmann, T. King, M. Busse, W. Effelsberg, and M. Fuchs. Scriptable sensor network based home-automation. In *EUC Workshops*, volume 4809 of *Lecture Notes in Computer Science*, pages 579-591. Springer, 2007.
- [27] M. Varchola, M. Drutarovsky, "Zigbee Based Home Automation Wireless Sensor Network" *Acta Electrotechnica et Informatica*, No. 4, Vol. 7, 2007, ISSN 1335-8243, Technical University of Košice, Slovak Republic.
- [28] Gauger, M., Minder, D., Marrón, P. J., Wacker, A., & Lachenmann, A. (2008). Prototyping sensor-actuator networks for home automation. In *Proceedings of the 3rd workshop on real-world wireless sensor networks (REALWSN 2008)*.
- [29] <http://www.pucc.jp/>
- [30] <http://www.crossbow.com>
- [31] <http://www.p3international.com/products/special/P4400/P4400-CE.html>
- [32] <https://www.wattsupmeters.com/secure/index.php>
- [33] <http://www.powermeterstore.com/>

- [34] Leland, E. S., White, R. M., and Wright, P. K. Energy scavenging power sources for household electrical monitoring. *PowerMEMS* (Dec. 2006)
- [35] Leland, E. S., White, R. M., and Wright, P. K. Design and Fabrication of a MEMS AC Electric Current Sensor. *Advances in Science and Technology* (Volume 54) pg 350-355
- [36] http://www.piezocryst.com/piezoelectric_sensors.php
- [37] <http://electronicdesign.com/article/components/piezoelectric-ceramics-science-meets-pottery18095.aspx>
- [38] S. Roundy and P.K. Wright, A piezoelectric vibration based generator for wireless electronics, *Proceedings of Smart Mater. Struct.*, vol. 13 (2004), pp. 1131–1142.
- [39] B. Wagner, W. Benecke, “Magnetically driven microactuators: Design considerations,” *Microsystem Technologies '90*, H. Reichl (Ed.), Springer-Verlag 1990
- [40] http://www.pge.com/includes/docs/pdfs/about/edusafety/training/pec/toolbox/tll/appnotes/using_current_as_proxy_for_power.pdf
- [41] http://www1.eere.energy.gov/buildings/appliance_standards/

[42] Mikhail Lisovich, Stephen Wicker. "Privacy Concerns in Upcoming Residential and Commercial Demand-Response Systems". 2008 Clemson University Power Systems Conference, Clemson University, March, 2008. Also presented at the TRUST 2008 Spring Conference, Berkeley CA, April 2008.

[43] C. Bennett, D. Highfill., "Networking AMI Smart Meters" *Energy 2030 Conference, 2008. ENERGY 2008. IEEE* In Energy 2030 Conference, 2008. ENERGY 2008. IEEE (2008), pp. 1-8.

[44] J. Zheng, M. J. Lee, and M. Anshel, "Toward secure low rate wireless personal area networks," *IEEE Transactions on Mobile Computing*, Vol. 5, No. 10, pp. 1361-1373, Oct. 2006.

[45] C. Perkins, E. Royer, S. Das RFC 3561 Ad hoc On-Demand Distance Vector (AODV) Routing

[46] <http://www.zigbee.org/>

[47] I. Chakeres, L. Klein-Berndt, AODVjr, AODV simplified, ACM SIGMOBILE Mobile Computing and Communications Review, July 2002, pp. 100–101.

[48] S. Tao, W. Wei, L. X. Dong, L. J. Wei, "AODVjr Routing Protocol with Multiple Feedback Policy for ZigBee Network" *In The 13th IEEE International Symposium on Consumer Electronics (ISCE2009)*

[49] <http://www.daintree.net/resources/index.php>

[50] J. Jun and M.L. Sichitiu, The nominal capacity of wireless mesh networks, *IEEE Wireless Communications* 10 (2003) (5), pp. 8–14.

[51] <http://www.howstuffworks.com>

[52] U.S. Dep't. of Energy, 109th Cong., Report on Section 1252 of the Energy Policy Act of 2005 (February 2006).

[53] [http://www.nist.gov/smartgrid/upload/InterimSmartGridRoadmapNIST Restructure.pdf](http://www.nist.gov/smartgrid/upload/InterimSmartGridRoadmapNISTRestructure.pdf)

[54] J. See, W. Carr, S. E. Collier. Real Time Distribution Analysis for Electric Utilities

[55] Bennett C., Wicker B. S., “Decreased Time Delay and Security Enhancement Recommendations for AMI Smart Meter Networks.” *2010 PES Innovative Smart Grid Technologies Conference.*, Gaithersburg, Maryland, January 19th-21st, 2010

[56] Chi Pan Chan, Soung Chang Liew, and An Chan, “Many-to-One Throughput Capacity of IEEE 802.11 Multihop Wireless Networks”, *IEEE Transactions on Mobile Computing*, Vol. 8, No. 4, April 2009, pp. 514-527

[57] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, pp. 388-404, Mar. 2000.

[58] Finn, G. Routing and addressing problems in large metropolitan-scale internetworks. ISI Res. Rep. ISI/RR-87-180, Univ. Southern California, Los Angeles, 1987.

[59] http://www.smartgridnews.com/artman/publish/News_Blogs_News/The-Coming-Smart-Grid-Data-Surge-1247.html

[60] E. J. Duarte-Melo, and M. Liu. Data-gathering wireless sensor networks: organization and capacity. Computer Networks (COMNET) Special Issue on Wireless Sensor Networks, Vol 43, Issue 4, pp. 519-537, November 2003.

[61] W.-Z. Song, F. Yuan, and R. LaHusen, "Time-Optimum Packet Scheduling for Many-to-One Routing in Wireless Sensor Networks," in *IEEE MASS*, 2006.

[62] C. T. Ee and R. Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks. In *Proc. SenSys'04*, Baltimore, MD, November 2004.

[63] Chenyang Lu , Brian M. Blum , Tarek F. Abdelzaher , John A. Stankovic , Tian He, RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks, Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'02), p.55, September 25-27, 2002

[64] B. Krishnamachari., "Networking Wireless Sensors" Cambridge University Press 2005

- [65] Tian He , John A. Stankovic , Chenyang Lu , Tarek Abdelzaher, SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks, Proceedings of the 23rd International Conference on Distributed Computing Systems, p.46, May 19-22, 2003
- [66] A. Acharya, A. Misra, S. Bensal, A label-switching packet forwarding architecture for multi-hop wireless LANs, in: M. Conti, D. Raychaudhuri (Eds.), Proceedings of the ACM Workshop on Mobile Multimedia (WoWMoM 2002), Atlanta, GA, September 28, 2002.
- [67] C. H. Yeh, *Ad hoc MPLS for virtual-connection-oriented mobile ad hoc networks*, IEEE 55th Vehicular Technology Conference, 2002. VTC Spring 2002., pp. 1101-1105, Vol. 3, 6-9 May 2002.
- [68] Ariza, Alfonso, Casilari, Eduardo and Cabrera, Alicia Triviño (2009), "An architecture for the implementation of Mesh Networks in OMNeT++", *OMNeT++ 2009: Proceedings of the 2nd International Workshop on OMNeT++ (hosted by SIMUTools 2009)*.
- [69] A. Ariza , E. Casilari, A. T. Cabrera, Implementation of a label based forwarding protocol for Ad-Hoc Networks
- [70] J. Choi, B. Y. Choi, S. Song, and K. H. Lee, NQAR- Network Quality Aware Routing in Wireless Sensor Networks, Lecture Notes in Computer Science, Wireless Algorithms, Systems, and Applications, pg 224-233

- [71] RFC 3031 Multiprotocol Label Switching Standard
- [72] IEEE 802.16 Standard
- [73] <http://www.wi-fiplanet.com/wimax/article.php/3065261>
- [74] S. Sharma, R. Gupta, Simulation Study of Blackhole Attack in The Mobile Ad Hoc Networks, In Journal of Engineering Science and Technology, Vol. 4, No. 2 (2009) 243 – 250
- [75] Mathew D. Penrose, On k-connectivity for a geometric random graph, Random Structures & Algorithms, v.15 n.2, p.145-164, Sept.1999
- [76] M.K. Marina and S.R. Das, “On-demand multipath distance vector routing in ad hoc networks,” *Proceedings of the International Conference for Network Procotols (ICNP)*, pp. 14–23, Nov. 2001.
- [77] A. Nasipuri, R.Castaneda, and S.R. Das, “Performance of multipath routing for on-demand protocols in mobile ad hoc networks,” *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 6, no. 4, pp. 339–349, 2001.
- [78] Feigenbaum, J., Johnson, A., Syverson, P.: A model for onion routing with provable anonymity. In: Financial Cryptography. LNCS, vol.4886. Springer, Heidelberg (2007)

[79] <http://www.freepatentsonline.com/4644320.html>

[80] <http://www.freepatentsonline.com/6402691.html>